Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ANGELINA CASSIA DE PEDRI

COMPARAÇÃO ENTRE MODELOS BANCO DE DADOS SQL E NoSQL

Assis/SP

2016

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ANGELINA CASSIA DE PEDRI

COMPARAÇÃO ENTRE MODELOS BANCO DE DADOS SQL E NoSQL

Projeto de pesquisa apresentado ao Curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial a obtenção do Certificado de Conclusão.

Orientanda: Angelina Cassia De Pedri

Orientador: Alex Sandro Romeo de Souza Poletto

Assis/SP

2016

FICHA CATALOGRÁFRICA

P371c PEDRI, Angelina Cássia de
Comparação entre modelos de bancos de dados SQL e NoSQL / Angelina Cassia De Pedri.—Assis, 2016.
44p.

Trabalho de Conclusão de Curso (Ciências da Computação). – Fundação Educacional do Município de Assis-FEMA

Orientador: Dr. Alex Sandro Romeo de Souza Poletto

1.SQL. 2. NoSQL

CDD: 005.74

COMPARAÇÃO ENTRE MODELOS BANCO DE DADOS SQL E NoSQL

ANGELINA CASSIA DE PEDRI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: _____

Analisador (1): _____

Assis/SP
2016

RESUMO

Este trabalho tem o intuito de despertar o interesse para ampliar o conhecimento sobre a tecnologia de armazenamento de dados não relacionais quando comparada a um modelo de banco de dados NoSQL que seria a tecnologia chamada *Big Data*. A tecnologia citada ainda é pouco explorada por ser um assunto relativamente novo nos dias atuais, porém pelas projeções analisadas possui um grande crescimento, pois o aumento do volume de dados gerados por redes sociais, dados meteorológicos, entre outras fontes vem crescendo a cada dia. Para armazenar este grande volume de dados com uma escalabilidade diferenciada são necessárias tecnologias que possam atender a esta demanda, que seriam os modelos de banco de dados NoSQL (Not Only SQL) que foi proposto para atender a estes requisitos gerenciando grandes volumes de dados não estruturados.

Palavras-Chave — SQL, NoSQL.

ABSTRACT

This dissertation aims to arouse interest to extend the knowledge about the technology of non-relational data store to a NoSQL database model it would be the technology called Big Data. The aforementioned technology is still little exploited for being a relatively new subject in the present day, though the projections analyzed has a large growth, because the increase in the volume of data generated by social networks, weather data, among other sources has been growing every day. To store this large volume of data with a differentiated scalability are necessary technologies that can meet this demand, which would be the NoSQL database models (Not Only SQL) that was proposed to meet these requirements managing large volumes of unstructured data.

Keywords: Keywords SQL; Keywords: NoSQL.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

1. INTRODUÇÃO

O termo *Big Data* está cada vez mais popular, porém, ainda não está bem claro o seu significado, a sua aplicabilidade e a sua finalidade. *Big Data* seria uma tecnologia voltada para banco de dados *NoSQL*, que armazenam dados não relacionais como imagens, dados meteorológicos, *posts* de redes sociais dentre outras informações (TAURION, 2013).

Para uma melhor compreensão, é essencial entender a definição das variáveis que compõem esta nova tecnologia que seriam os 3V's: Volume + Variedade + Velocidade.

Big Data vem chamando atenção pela acelerada escala em que volumes cada vez maiores de dados são criados pela sociedade. No entanto, existem muitas dúvidas de como tangibilizar o conceito, ou seja, como sair do conceitual e criar soluções de negócio que mineralizem esta massa de dados, já que a cada dia são gerados dezenas de *petabytes* de dados, em uma escala real e não mais imaginária e futurista.

Um banco de dados com a estrutura do *Big Data* permite a utilização de diversas tecnologias de gerenciamento para organizar estes dados que são gerados massivamente, como, aplicação de *Data Warehouse*, *Data Mart*, *Data Mining*, *Web Data Mining*, BI (*Business Intelligence*), *Cloud Computing* e aplicação do algoritmo de redes neurais.

No entanto, as tecnologias atuais de gerenciamento de dados, como o modelo relacional proposto por Edgar F. Coode em 1969, não são mais adequadas para suportar os dados com a estrutura do *Big Data*. Para tratar dados na escala de Volume, Variedade e Velocidade do *Big Data*, são necessários outros modelos mais apropriados como os sistemas gerenciadores de bancos de dados *NoSQL*, projetados para tratar imensos volumes de dados estruturados e não estruturados.

Existem diversos modelos como sistemas colunares como o *Big Table*, usados internamente pelo Google ou o modelo *Key/value* como *DynamoDB* da Amazon e o *"Document Database"* baseado no conceito proposto pelo Lotus Notes da IBM e aplicado em softwares como *MongoDB*, BARASUOL (2012).

1.1 OBJETIVOS

Os estudos citados têm por finalidade explorar as tecnologias que envolvem o processo de armazenamento de dados, uma forma que abrange técnicas, métodos e etapas para aplicação, no sentido de auxiliar os especialistas na identificação de problemas com armazenamento de dados no dia-a-dia e a exemplificação de um banco de dados NoSQL.

Esta pesquisa leva a sintetizar os principais conceitos relacionados ao *Big Data* e tecnologias que o abrangem, tais como Banco de Dados NoSQL juntamente com a tecnologia do MongoDB, um banco de dados não relacional orientado a documentos. Espera-se que este trabalho apresente como resultado uma base introdutória para conhecimento do tema pesquisado, podendo ser utilizado para familiarização do assunto.

Observando o interesse e as dificuldades para a compreensão do *Big Data*, a existência de um trabalho que aborde e analise o tema como este pode ser relevante e justo pelo fato de auxiliar na introdução de profissionais ou leigos ao tema ampliando sua grade de conhecimento voltada para o armazenamento de dados.

1.2 MOTIVAÇÃO

Tendo em vista a diversidade de alternativas citadas no início deste capítulo é requerido conhecimento e estudo sobre as tecnologias existentes com o intuito de aprimorá-las para que seja possível tratar os dados não estruturados de uma maneira eficaz, considerando o Volume, Variedade e Velocidade com que esses dados são gerados e acessados.

Para tratar dados na escala dos 3 V’s do *Big Data*, são necessários outros modelos mais apropriados como os sistemas gerenciadores de bancos de dados NoSQL, projetados para tratar imensos volumes de dados estruturados e não estruturados.

O intuito deste trabalho é de levar estas novas tecnologias relacionadas com armazenamento de dados e melhoria das consultas a estudantes, profissionais da área de TI e demais interessados.

Esta justificativa leva adiante este estudo mantendo o foco da pesquisa em novas tecnologias de armazenamento de banco de dados NoSQL, para contribuir com melhorias voltadas a esta área.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 5 capítulos, sendo esta Introdução o primeiro.

No segundo capítulo serão apresentados os conceitos relacionados a Sistemas Gerenciadores de Banco de Dados (SGBD) SQL e NoSQL.

A Proposta de Trabalho será apresentada no terceiro capítulo.

O quarto capítulo conterá o Estudo de Caso apresentando as etapas utilizadas para chegar ao objetivo proposto.

E no quinto capítulo serão citadas as Considerações Finais.

Por último, serão relacionadas as Referências Bibliográficas.

2. SISTEMA GERENCIADOR DE BANCO DE DADOS NoSQL

Os SGBD NoSQL (*Not Only SQL*) levam este nome por sua maneira diferenciada de escrever as *queries*, onde cada modelo de banco de dados possui uma forma de expressão.
Hoje, a grande motivação para o estudo relacionado aos bancos de dados NoSQL parte da resolução dos problemas ligados a escalabilidade, onde pode se tornar muito complexo e custoso manter um banco de dados tradicional, em SQL (*Structured Query Language*).
Como uma das principais características dos SGBD NoSQL destacasse a ausência de um *schema* onde não se identifica a integridade e os relacionamentos, ou seja, não fazendo uso da álgebra relacional como os bancos de dados tradicionais.

2.1 DEFINIÇÃO DE UM BANCO DE DADOS SQL

Conforme descrito por LOSCIO, *et al* 2014, os SGBD relacionais surgiram no início dos anos 70 onde foi uma solução em armazenamento de dados para o setor comercial e gerenciamento de dados convencionais, como dados estruturados gerados por sistemas comuns.
O modelo relacional tem como conceito o relacionamento entre tabela, atributo e tupla, conforme o exemplo apresentado na Figura 1 (tabela FUNCIONARIOS).

FUNCIONARIOS		
Nome	Cargo	Salario
Marcos Silva	Mecanico	2.000,00
Joao Oliveira	Motorista	1.500,00
Bruno Santos	Soldador	2.100,00
Pedro Ferreira	Guarda	1.100,00

Figura 1: Exemplo de uma tabela em um SGBD relacional

Fonte: Autoria própria

Os SGBD's relacionais têm como principais características o controle de concorrência, segurança, recuperação de falhas, gerenciamento do mecanismo de armazenamento de dados de controle de restrições e integridade, (LOSCIO, *et al*, 2014) e um outro ponto que

se faz necessariamente importante em um BD SQL são as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade).

Entre as características citadas sobre um banco de dados relacional, o mesmo destaca-se também com sua padronização e sua base formal com a simplicidade da linguagem SQL, padrão para manipulação de dados relacionais.

2.2 DEFINIÇÃO DE UM BANCO DE DADOS NoSQL

Os SGBD's NoSQL possuem como característica marcante a não utilização da álgebra relacional que é utilizada nos bancos de dados SQL e não possuem linguagem nativa.

Os mesmos são compostos de uma estrutura simples, sem esquemas e por possuírem a capacidade de armazenar dados com o crescimento exponencial, possuem grande desempenho no armazenamento dos dados provenientes da Web (redes sociais, e-mails, dados meteorológicos), (PRADO, et al, 2014).

Faz-se notável também a simplicidade na manipulação das informações, pois não possuem a estrutura relacional voltada a replicação.

Um dos principais conceitos relacionados ao armazenamento de dados em um banco NoSQL é o tipo de dados que o mesmo está preparado para receber, como dados estruturados, *semiestruturado*s e não estruturados, onde respectivamente o primeiro modelo de dados possui uma estrutura formal, como informações de um sistema comum, o segundo não possui estrutura rígida, mas pode possuir *tags* XML ou JSON e o terceiro não possui nenhuma forma de estrutura, onde se encaixariam os dados provenientes das redes sociais, áudio, vídeo, entre outros.

Estes dados recebem o tratamento do modelo *key-value*, onde registros são armazenados em pares de chave e valor, assim salvam os dados com o nome sugerido formado por uma chave e um conjunto de entradas que são associadas a um valor e o mesmo pode ser binário ou *string* que vão sendo salvos de forma livre (schema-free) (STEPPAT, 2009).

Porém, existem características mais especificas relacionadas a cada tipo de banco de dados NoSQL que serão apresentadas a seguir, tornando-os mais adequados para o tratamento do grande volume de dados que são armazenados.

2.2.1 Escalabilidade horizontal

Não possuem controle bloqueios, onde torna-se prático para o gerenciamento de grandes volumes de dados.

2.2.2 Ausência de esquema (*Schema-Free*)

Um BD NoSQL pode ter a ausência completa ou quase toda de esquema, ou seja, pode ser semiestruturado ou não estruturado os dados a serem armazenados. Assim, é classificado com um banco de dados livre esquema estruturado, porém não tem a integridade como propriedade.

2.2.3 Suporte a replicação

Outra maneira de obter a escalabilidade seria por meio da replicação, onde diminui o tempo na recuperação das informações.

2.3 MODELOS DE BANCOS DE DADOS NoSQL

A seguir serão apresentadas algumas vantagens e limitações e os modelos de NoSQL mais conhecidos e utilizados.

Como vantagens pode-se citar que o NoSQL seria uma base de dados muito flexível, pois não fica presa a um esquema relacional onde possui a facilidade de inserção de novos dados, possuindo um número menor de manutenções comparado ao SGBD SQL, tendo em vista também a redução do custo das *queries*.

Quanto as limitações, não seria interessante armazenar dados relativos em um SGBD NoSQL, pois sua forma de armazenamento seria feita como uma lista no BD SQL e no NoSQL pode ser que estas informações não compreendam a mesma lista, onde não seria possível contar o número de utilizadores, por exemplo em uma rede social, como as

informações ficam dispersas seria um tanto complexo contabilizar quantos *likes, posts* um determinado assunto obteve (STEPPAT, 2009).

2.3.1 Key/Value (chave/valor)

Este modelo é considerado simples comparando a visualização do banco de dados a uma tabela de *hash*, composto por um conjunto de chaves associadas a um valor podendo ser *string* ou binário (PRADO, 2014).

Conforme a representação no banco de dados na Figura 2 é exemplificada a estrutura chave/valor como um exemplo a utilização o SGBD NoSQL.

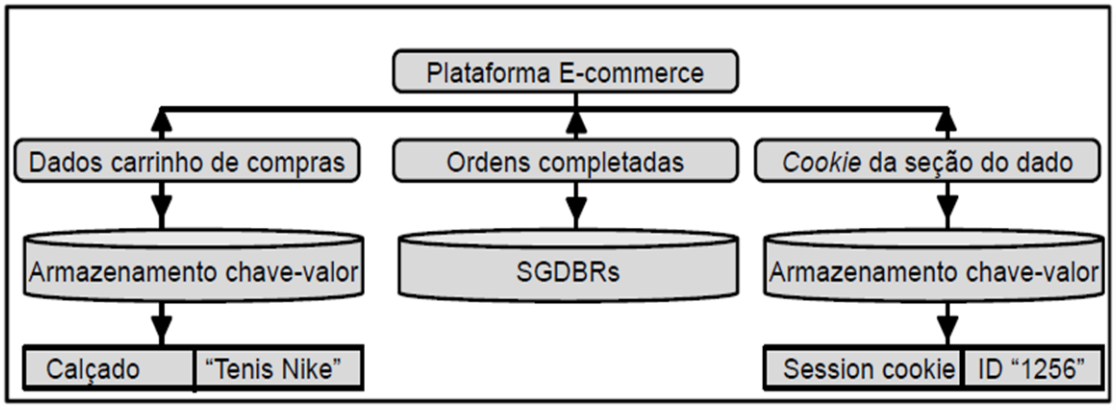

Figura 2. Banco de dados NoSQL orientado a chave/valor

Fonte: adaptado de CRITÉRIOS DE SELEÇÃO DE SGBD NOSQL EM ORGANIZAÇÕES BRASILEIRAS, Prado 2014

Como exemplo da utilização deste modelo de banco de dados NoSQL pode-se citar o *Dynamon*, desenvolvido pela Amazon. Uma das características deste BD é a possibilidade do particionamento, replicação e versionamento dos dados.

2.3.2. Orientado a colunas

O modelo orientado a colunas é mais complexo do que o modelo chave-valor, mudando a representação de chave/valor para registros ou tuplas para orientação de tuplas ou colunas. Este modelo de banco de dados NoSQL teve origem com o *BigTable* da Google (STEPPAT, 2009).

Na Figura 3 segue a exemplificação de um banco de dados no modelo orientado a colunas.

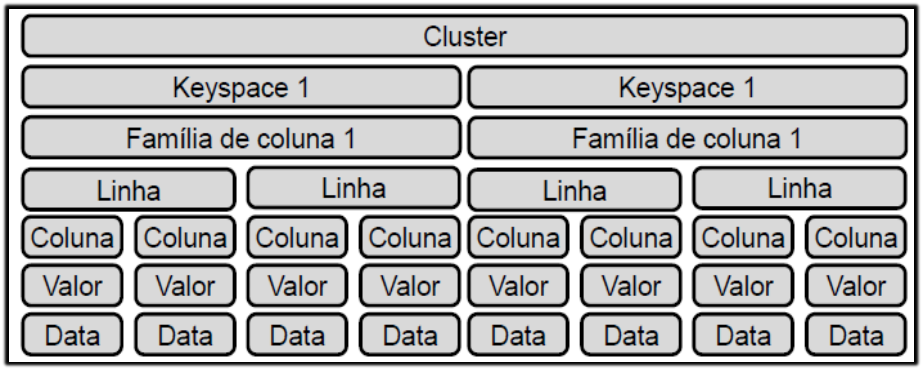

Figura 3. Banco de dados NoSQL orientado a colunas

Fonte: adaptado de CRITÉRIOS DE SELEÇÃO DE SGBD NOSQL EM ORGANIZAÇÕES BRASILEIRAS, Prado 2014

2.3.3. Orientado a documentos

Pode-se dizer que este modelo é uma coleção de documentos, onde um documento é um objeto com um identificador único e um conjunto de campos, que podem ser *strings*, lista ou documentos ordenados (LÓSCIO, *et al*, 2014).

O conjunto de campos tem a orientação como o modelo chave-valor, sendo que no orientado a documentos tem-se um conjunto de documentos e em cada documento um conjunto de campos (chaves) e o valor deste campo.

Este modelo não depende de um esquema rígido ou uma estrutura fixa como os bancos relacionais.

Na Figura 4 segue uma exemplificação de um banco de dados NoSQL orientado a documentos no qual os dados são divididos em *shards* (fragmento de banco de dados) e cada um conta com sua própria replicação.

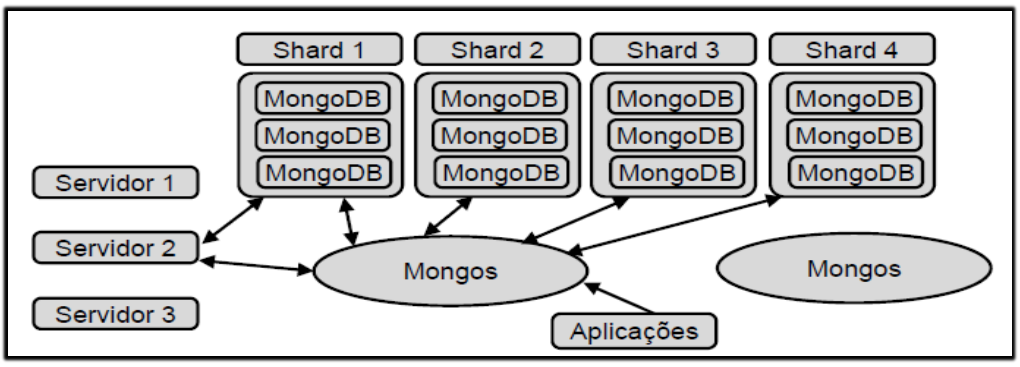

Figura 4. Banco de dados NoSQL orientado a documentos
Fonte: adaptado de CRITÉRIOS DE SELEÇÃO DE SGBD NOSQL EM ORGANIZAÇÕES BRASILEIRAS, Prado 2014

As soluções que utilizam este modelo de banco de dados são o *CouchDB* e o *MongoDB*. O *CouchDB* foi implementado em Java utilizando o formato JSON já o *MongoDB* foi em C++.

2.3.4. Orientado a grafos

Este modelo é composto por três componentes básicos: os nós (são os vértices do grafo), os relacionamentos (são as arestas) e as propriedades (ou atributos) dos nós e relacionamentos (LÓSCIO, *et al*, 2014).

Assim, o banco de dados é visto como um multigrafo onde cada par de nós pode ser conectado por mais de uma aresta.

A seguir, na Figura 5, tem-se 3 indivíduos, representando os nós do grafo, que estão conectados as cidades, que representam as arestas onde torna-se mais simples a representação para entendimento, no qual a seta azul representa a cidade para qual a pessoa viajou e a linha vermelha onde morou.

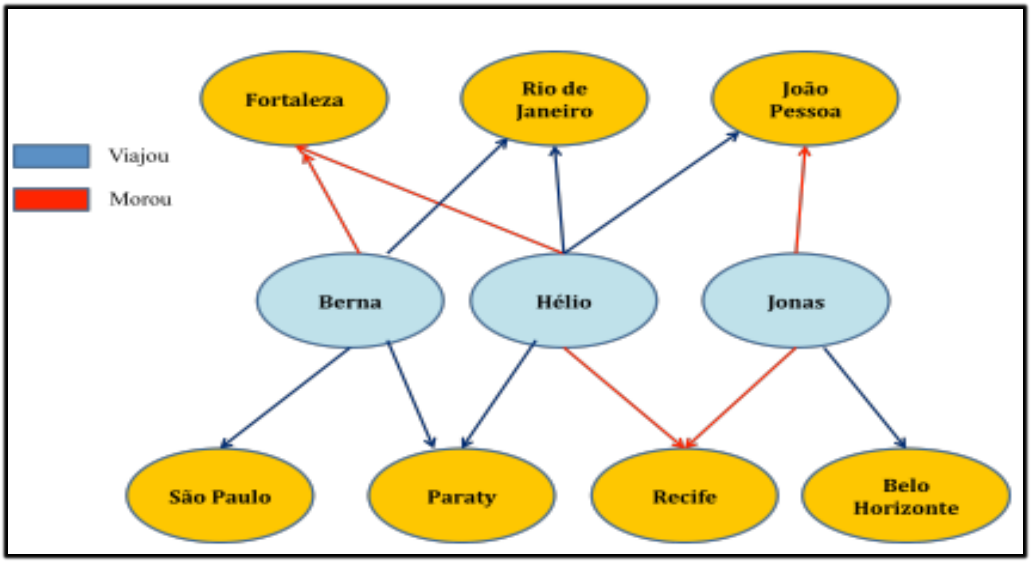

Figura 5. Banco de dados NoSQL orientado a grafos
Fonte: adaptado de NoSQL no desenvolvimento de aplicações Web colaborativas

Como exemplos de bancos de dados orientados a grafos pode-se citar o Neo4j14, AllegroGraph15 e Virtuoso16. O Neo4j é *open-source* e foi implementado em Java e

AllegroGraph e o Virtuoso são baseados no modelo RDF (*Resource Description Framework*) modelo padrão para representação de dados na Web.

2.3.5 MongoDB

Nos últimos anos vem sendo observado o grande interesse pelos SGBD NoSQL, que diferem do modelo relacional SQL, como citado no Capítulo 2, é um dos projetos que vem sendo mais notado no momento é o MongoDB, que teve início em 2007, porém só foi concluído em 2009 quando foi lançada a sua primeira versão (MEDEIROS, 2016).

O MongoDB é um SGBD orientado a documentos e de software livre que armazena dados em coleções de documentos semelhantes a JSON (*JavaScript Object Notation* - Notação de Objetos JavaScript).

O que distingue o MongoDB de outros bancos de dados NoSQL é a sua linguagem de consulta baseada em documentos, que torna a transição de um banco de dados relacional para este BD mais fácil, porque as consultas são convertidas com bastante facilidade.

Um banco de dados Orientado a Documentos possui a característica de trazer todas as informações relevantes em um único documento, que não possui esquemas e é composto pelo UUID (Identificadores Únicos Universais), possibilitando a consulta através de métodos mais avançados como, por exemplo, o Map/Reduce (agrupa e filtra) que permite também a redundância e a inconsistência (MEDEIROS, 2016).

Os bancos de dados orientados a documentos diferem significativamente dos bancos de dados relacionais, visto que não possuem o esquema de armazenamento como tabelas e os dados são armazenados em documentos vagamente definidos.

Por exemplo, nos SGBD tradicionais, se for necessário adicionar uma coluna a uma tabela, altera toda a estrutura da mesma colocando esta coluna relacionada a todos os outros registros existentes na tabela. Esta situação ocorre pelo design limitado dos SGBD tradicionais.

Já em SGBD NoSQL orientado a documentos, é possível acrescentar novos atributos aos documentos individualmente sem que altere os demais; devido ao design de bancos de dados orientados a documentos não possuir esquema.

Com essas informações já é possível identificar a diferença notável entre o banco de dados relacional que possui um esquema para organização dos dados com o banco de dados não relacional que é livre deste esquema onde, no SQL a representação dos registros é feita

utilizando a abordagem linha/coluna e no caso deste do NoSQL os registros são armazenados em documentos formando uma “coleção”, (LENNON, 2011).

O MongoDB possui o código-fonte implementado em C++ e aberto licenciado pela GNU, possibilitando alta performance, multiplataforma e é formado por um conjunto de aplicativos armazenando os dados dentro de documentos semelhantes a JSON (usando BSON — uma versão binária de JSON), que retém os dados usando pares de chave/valor (LENNON, 2011).

Este banco de dados, por ser baseado em documentos, não possui transações tornando a consulta às informações mais simples, facilitando também a escrita da query no modelo BSON.

Como exemplo, pode-se utilizar o conceito da tabela CONTATOS que é populada pelos nomes dos contatos onde possui os atributos ‘nome’, ‘e-mail’ e ‘mensagem’.

Na Figura 6 segue a forma que os contatos ficam armazenados no banco de dados tradicional (SQL), conforme a seguir:

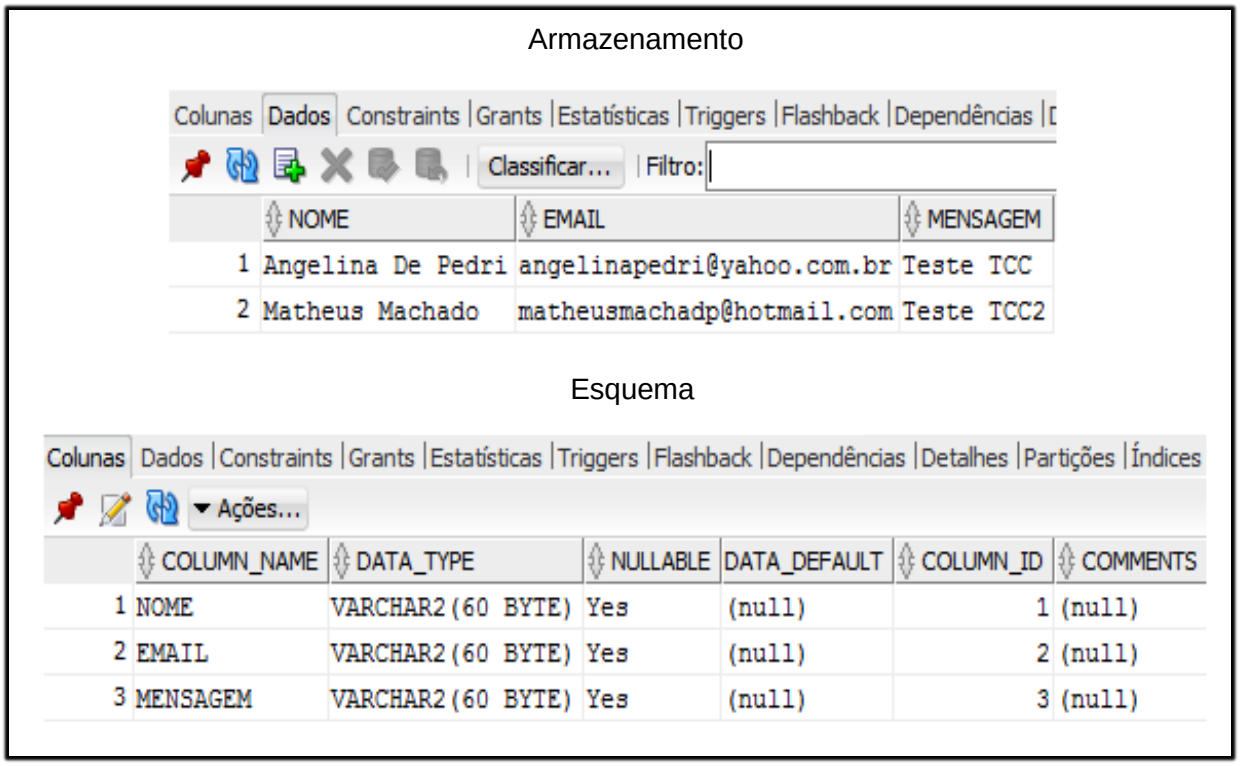

Figura 6: Demonstração do armazenamento e o esquema em um banco de dados tradicional SQL

Fonte: Autoria própria

Na Figura 7 é apresentado um exemplo de como fica esta consulta em um banco SQL:

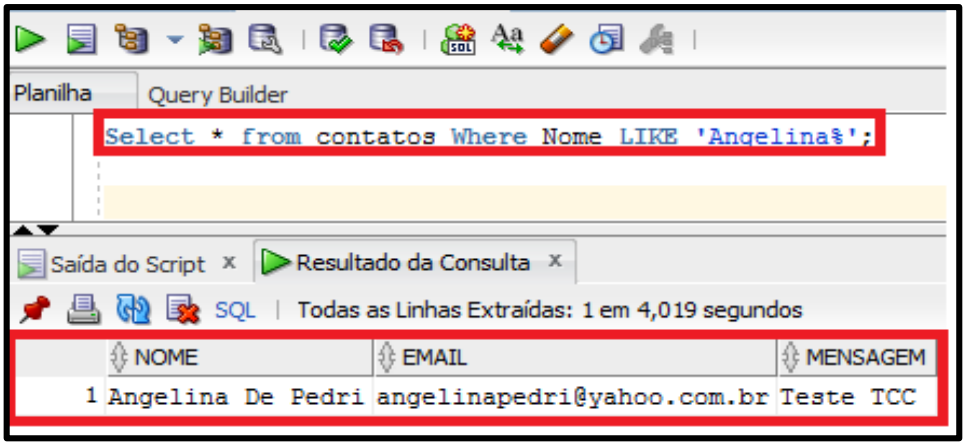

Figura 7: Demonstração de uma consulta no BD SQL

Fonte: Autoria própria

Segue agora uma demonstração utilizando o mesmo conceito acima da tabela CONTATOS, que possui apenas uma coleção (contatos), no qual está representado por uma lista de documentos.

Os dados armazenados são representados conforme Figura 8:

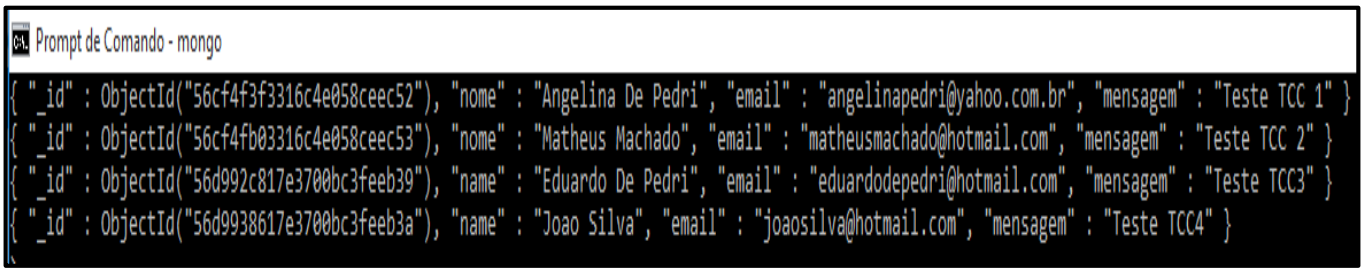

Figura 8: Demonstração de como inserir/armazenar dados em uma coleção no BD NoSQL MongoDB

Fonte: Autoria própria

No MongoDB, Orientado a Documentos a consulta fica conforme a seguir, Figura 9.

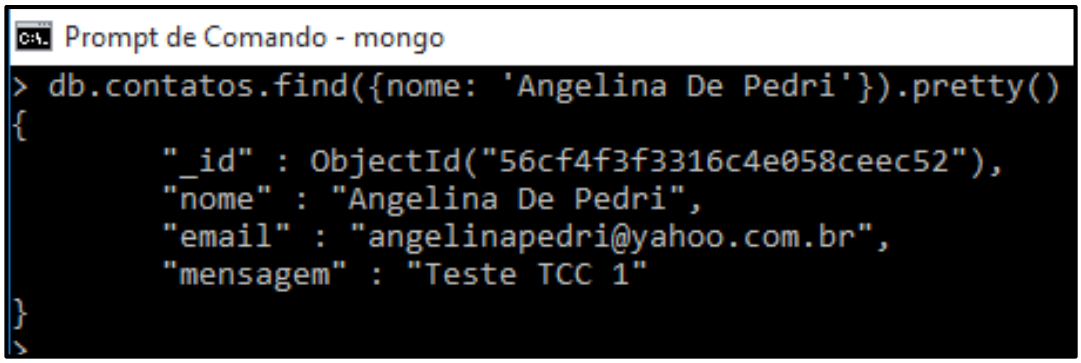

Figura 9: Demonstração de uma consulta no BD NoSQL MongoDB

Fonte: Autoria própria

Os itens apresentados a seguir, sobre os recursos do MongoDB, são baseados no artigo de LENNON,2011.

O MongoDB possui mais recursos além do armazenamento básico de chaves/valores, conforme segue:

- Binários oficiais disponíveis para Windows®, Mac OS X, Linux® e Solaris, distribuição de código de origem disponível para autoconstrução;
- Drivers oficiais disponíveis para C, C#, C++, Haskell, Java™, JavaScript, Perl, PHP, Python, Ruby e Scala, com uma ampla variedade de drivers suportados pelas comunidades disponíveis para outras linguagens;
- Suporte a expressões regulares em consultas;
- Os resultados da consulta do MongoDB são armazenados em cursores que fornecem funções para filtragem, agregação e classificação como: limit(), skip(), sort(), count(), distintic() e group;
- Map/Reduce -- implementação para agregação avançada (agrupa e filtra);
- Replicação mestre/escravo similar ao MySQL;
- Escalada horizontal com *auto-sharding*.

Sobre a escala com *auto-sharding*, muito importante no armazenamento de dados do MongoDB, pode-se dizer que é altamente escalável já que possui a capacidade de armazenar altos volumes de dados não estruturados fazendo com que sua manutenção se torne mais fácil.

As implementações do MongoDB são escaladas horizontalmente utilizando o mecanismo *auto-sharding*, em que permite a escala de uma configuração de milhares de nós com o balanceamento de carga automático.

Quanto a replicação, o MongoDB fornece recursos com uma configuração mestre-escravo (semelhante ao MySQL), para fins de *failover* e redundância, garantindo alto nível de consistência entre os nós. Como alternativa, o MongoDB pode usar conjuntos de réplicas para definir um nó como principal, e a qualquer momento com outro nó assumindo, em caso de falhas, o nó principal.

Já os GridFS, auxiliam no armazenamento de grandes quantidades de dados que os documentos em BSON são capazes de armazenar, que é no máximo 4MB, e este recurso divide em pedaços menores estas informações em pequenos documentos.

2.4 BIG DATA E SEUS FATORES

Big Data é o termo utilizado atualmente para nomear a grande quantidade de dados armazenados em servidores vindos de diversas fontes de dados, como mídias sociais (*Twitter, Facebook, e-mails*), sensores, IoT (Internet of Things), dados meteorológicos, agricultura de precisão entre outras.

Estes dados não tinham o aproveitamento adequado para extração das informações, e dessa maneira acabavam perdendo seu valor. Porém, hoje com a solução NoSQL é possível tirar grande aproveito dos mesmos, pois o modelo relacional não poderia ser apropriado para armazenar dados não estruturados (TAURION, 2013).

Este segmento em busca por mais informações sobre esta tecnologia existe há mais de cinco anos no mundo, sendo que no Brasil começa a aparecer agora com maior destaque (DOURADO, 2013). Nesse período, no mundo, são visíveis os grandes avanços nas empresas que utilizam dessas tecnologias. É o caso de empresas como Google, Yahoo e Apple, dentre outras, ou seja, não se trata apenas do armazenamento de grandes volumes de dados inclui-se também o processamento intensivo dessas informações.

Esta solução para o armazenamento destes dados, vem sendo chamada de *Big Data*, que para relembrar este termo o mesmo é definido como 3 V's, Volume+Variedade+Velocidade (TAURION, 2013) onde:

Volume: representa as informações geradas pelos sistemas transacionais somadas aos dados gerados pelos sensores, câmeras, mídias sociais, via *smartphones*, *tablets*, entre outros meios de comunicação utilizados.

Variedade: define a forma que estes dados são apresentados, já que podem ser estruturados, semiestruturados ou não estruturados tais como fotos, *e-mails*, *logs*, *posts* e demais.

Velocidade: representa a agilidade da resposta quase que em tempo real para agir no próprio evento gerador das informações tratando um volume massivo de dados na casa de terabytes, zetabytes e petabytes.

Como pode-se observar a cada dia que passa o volume de dados vem aumentando, tanto de redes sociais como das pequenas e grandes empresas, os dados meteorológicos, da agricultura de precisão entre outros.

Para esses dados "desordenados" existem os Bancos de Dados NoSQL que auxiliam na organização e armazenamento, ou seja, bancos de dados não relacionais que tem a

capacidade de armazenar os dados estruturados, semiestruturados e não estruturados em melhor escala, utilizando os modelos disponíveis, conforme apresentado no Capítulo 2.

O modelo da geração destes dados e a forma de armazenamento sem estrutura mais a solução de banco de dados NoSQL é que formam o *Big Data*, um modelo de banco de dados não relacional.

Hoje em dia, é possível observar os benefícios de negócios tangíveis e pragmáticos com o uso do *Big Data*, seja para aumentar a taxa de conversão para reservas, diminuir custos de operação, impulsionar receita ou elevar a satisfação do consumidor (TAURION, 2013).

Por exemplo, para uma empresa aérea, com a tecnologia em tempo real e o armazenamento de grande quantidade de dados que o Big Data possibilita, é possível ter o perfil de cada passageiro, sabendo o destino mais procurado, tipo de assento, época de viagens, e demais informações, utilizando as características de cada um no momento da compra das passagens.

Assim esta empresa pode encaminhar pacotes promocionais e demais vantagens que são direcionadas exatamente para o perfil do cliente selecionado, conquistando mais clientes de acordo com as escolhas personalizadas e aumentando também os lucros.

Todos estes dados possuem um papel muito importante para as empresas e usuários que dependem das informações que são geradas através dos mesmos e também agilidade no retorno em um momento de consulta a estas informações contando também com a relevância deste retorno para tomadas de decisão.

Atualmente o *Big Data* encaixa-se na quarta posição entre as quatro principais plataformas de TI levando em consideração a sua grande importância entre o mercado econômico onde é utilizado para extrair valores de um grande volume de dados que serão utilizados para tomadas de decisão e projeções futuras (PINA, 2015).

As estimativas para o *Big Data* vêm crescendo juntamente com o volume de dados gerados e com a evolução das empresas.

Como exemplo, no ano de 2013, o *Big Data* na América Latina foi estimado em US$ 551 mi e no final de 2015 foi projetado para alcançar US$ 880 mi onde respectivamente cada ramo contribui para esta estimativa em seu percentual como em 2014, em que a Manufatura contribuiu com 27%, Finanças 21% e Comércio 12% dentre outros setores como

Comunicação e Mídia, Serviços Profissionais e Governamentais somaram os 40% (PINA, 2015).

De acordo com PINA, 2015, pode-se classificar quatro fatores que vem contribuindo para o aumento da estimativa do *Big Data* juntamente com a evolução das empresas.

O fator com maior relevância seria o IoT, tecnologia que as empresas vêm explorando para o aumento do volume da geração de dados on-line para servir na tomada de decisões em tempo real.

Em seguida tem-se o *Costumer Experience*, onde as empresas conseguem ter uma visão mais ampla das preferências de seus clientes, com base na análise adquirida no histórico de consulta de compras do determinado cliente.

O próximo seriam as Redes Sociais que a cada momento geram quantidades cada vez maiores de dados não estruturados que irão precisar realizar o relacionamento com dados estruturados em que o *Big Data* auxilia nesta manutenção mantendo o máximo de aproveitamento dos mesmos.

E o último ponto entre este *“hanking”*, é o *Data Warehouse*, tecnologia em armazenamento de dados que vendo crescendo na utilização das empresas juntamente com sua evolução, no qual a ideia desta tecnologia é integrar os dados internos e externos das empresas criando uma única estrutura permitindo uma melhor análise destes dados. Quando esta estrutura já está formada é aplicada a tecnologia de OLAP (*On-line Analytical Processing*) e *Data Mining* para a análise e mineração das informações que serão obtidas através dos dados coletados (PINA, 2015).

3. PROPOSTA DE TRABALHO

Para a execução deste trabalho, inicialmente foram apresentados nos capítulos anteriores os conhecimentos adquiridos com o levantamento bibliográfico referente aos temas abordados como *Big Data*, SGBD *NoSQL* e sobre o modelo orientado a documento, (MongoDB). Será apresentado o modelo do banco de dados relacional *SQL* através da ferrramenta *DBDesigner* 4.5.0, e em seguida serão criadas tabelas no *SQL Developer* 4.0.1 através de comandos composto pela álgebra relacional de um banco de dados tradicional SQL utilizando como SGBD *Oracle* 11 g *Enterprise Edition Release* 11.2.0.1.0.
Na Figura 10 é apresentada a proposta de trabalho, ilustrando de forma geral o contexto da mesma contendo as etapas e suas ramificações.

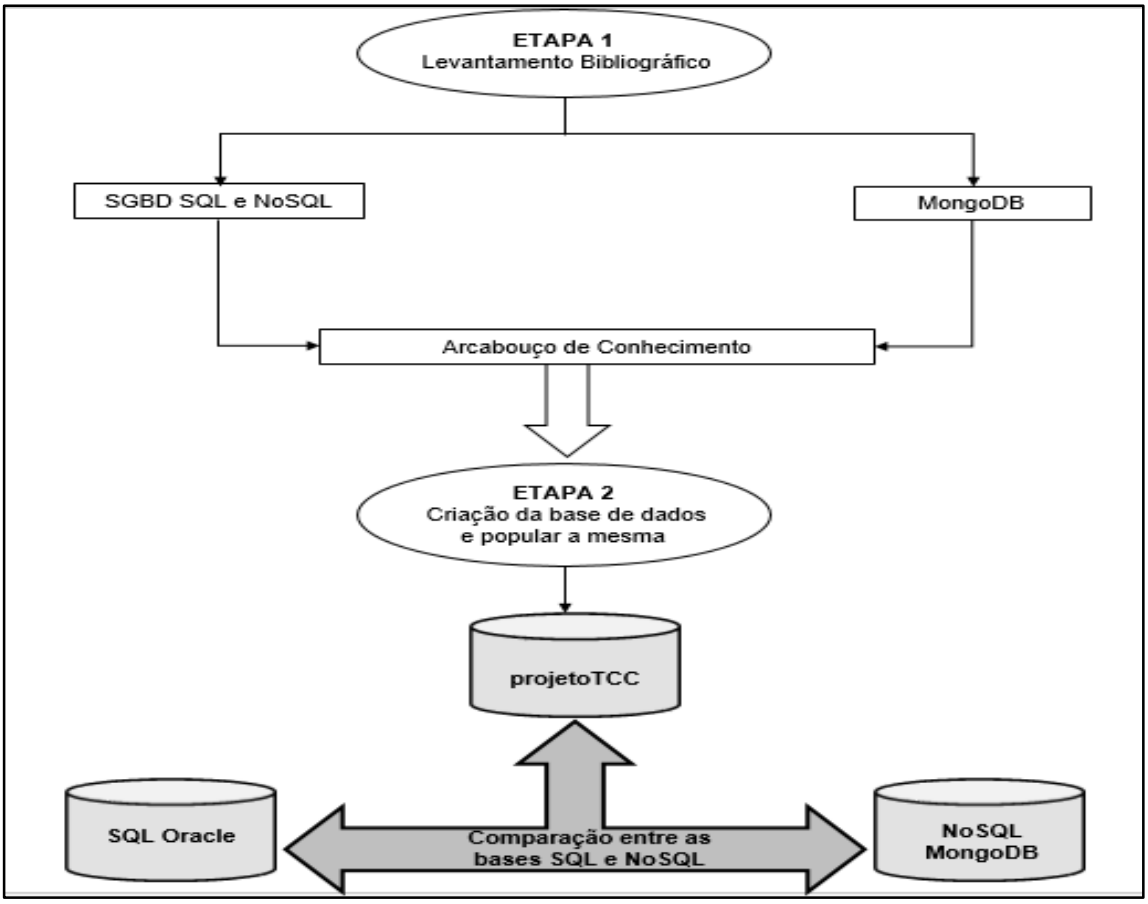

Figura 10: Proposta de trabalho

Posteriormente, será a base de dados do SGBD MongoDB 3.2.3 será povoada utilizando o conceito da linguagem BSON padrão para este BD e a estrutura de armazenamento será apresenta pelo *Shell* de comandos do MongoDB.

Com intuito de comparação entre *NoSQL* e *SQL* será feita uma apresentação onde ambos os SGBDS serão executados afim da apresentação citada.

3.1 ETAPA 1: LEVANTAMENTO BIBLIOGRÁFICO

A primeira etapa que compõem a proposta de trabalho é estruturada por um levantamento bibliográfico afim de adquirir mais conhecimentos sobre os temas abordados nos Capítulos bem como compreender a diferença entre um banco de dados relacional SQL e um não relacional NoSQL, tanto no armazenamento de dados como também na utilização dos comandos que são utilizados em ambos.

Hoje a utilização destes dois modelos de banco de dados faz parte do dia a dia, como por exemplo o banco de dados relacional armazena dados estruturados de tabelas que possuem relacionamentos entre tuplas e atributos, como uma tabela de registro de clientes por exemplo, já o banco de dados não relacional pode ser utilizado a fim de armazenar os dados não estruturados como mensagens, *posts*, e-mails, *likes* entre outros.

3.2 ETAPA 2: CRIAÇÃO DAS BASES DE DADOS E POVOAMENTO

Na segunda etapa cabe a demonstração das bases de dados SQL (relacional) e NoSQL (não relacional).

Para esta apresentação serão criadas duas bases de dados composta pelas mesmas informações, uma no SGBD Oracle e outra no MongoDB. Com esta demonstração pretende-se comparar os dois modelos quanto a forma de armazenamento dos mesmos e sua disposição para execução dos comandos das linguagens DML e DDL.

Para esta demonstração pretende-se utilizar algumas *querys* que irão ser compostas pelos comandos básicos que compõem a estrutura da instrução DML como INSERT, UPDATE e DELETE juntamente com a recuperação de dados através da instrução SELECT, com a finalidade da comparação entre os SGBD SQL e NoSQL.

4. ESTUDO DE CASO

Neste capitulo será avaliada a teoria aplicada na prática referente a Etapa 2 da proposta de trabalho.

4.1 MODELO DE DADOS – PROJETO TCC (SQL)

O modelo apresentado na Figura 11, contém três tabelas de dados do Projeto TCC, representando um relacionamento de uma base dados SQL – relacional. Como pode ser observado, em um modelo relacional existem chaves primárias que são essenciais para o relacionamento com outras tabelas, onde serão agregadas como chaves estrangeiras mantendo a integridade e a composição da base de dados.

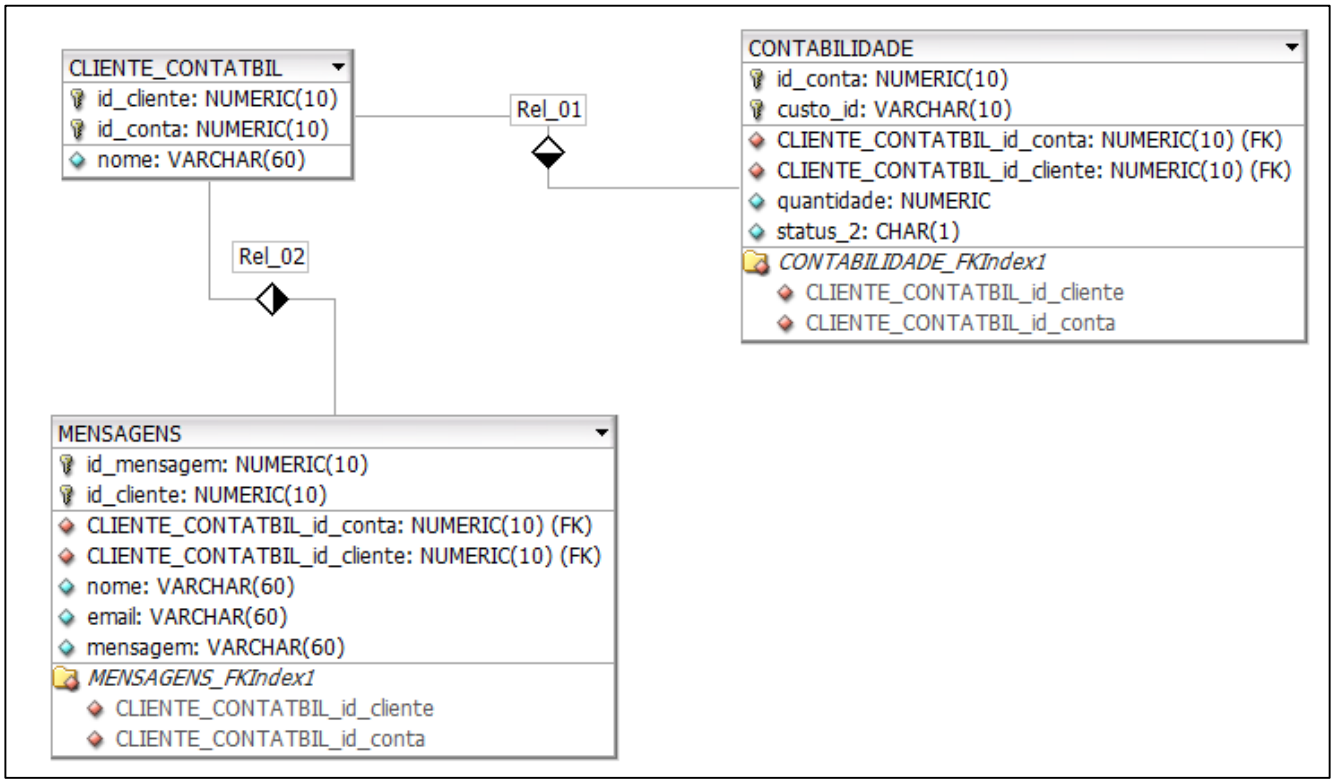

Figura 11: Modelo da base de dados Projeto TCC – SQL

Fonte: Autoria própria

4.1.1 Criação da base dados

A Figura 12, representa a SQL utilizada para criação da Base de Dados Relacional.
Para a criação da mesma foi utiliza a instrução DDL *Create* seguida da instrução *Database* (base de dados) e nome da base que será criada, permitindo ao usuário realizar a criação de uma base de dados.

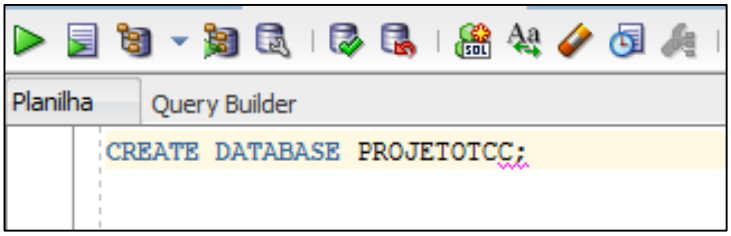

Figura 12: Exemplo de SQL utilizada para criação da base dados relacional

Fonte: Autoria própria

4.1.2 Criação das tabelas e inserção de dados

Na Figura 13 está representada a SQL de exemplo para criação das tabelas de dados relacionais e na Figura 16 a SQL para a carga de dados.
Ao observar a SQL da Figura 13 nota-se que para criar a tabela CONTABILIDADE foi utilizada a instrução SQL-DDL *Create Table*. A declaração do atributo ID como *not null*, estabelecer o tipo de dados para cada atributo e ao final da *query* atribuir o atributo ID como chave primária, já que na base relacional não é feita a criação do atributo ID automaticamente, ela é definida pelo usuário.

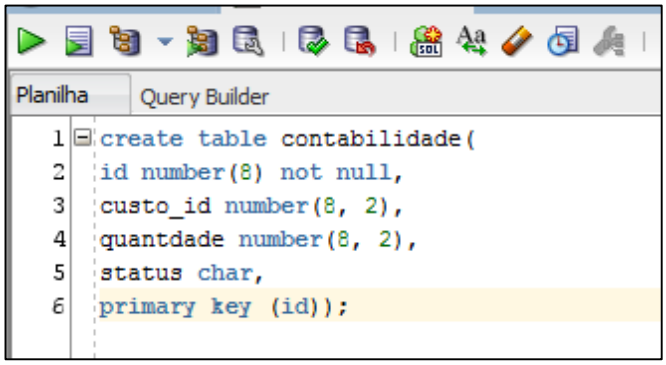

Figura 13: Exemplo de SQL executada no banco de dados relacional para criação das tabelas

Fonte: Autoria própria

Na Figura 14, referente a carga de dados, é possível verificar a utilização da instrução SQL-DML *Insert*, da instrução *into* que indica que os dados serão inseridos na tabela escolhida, juntamente com a função *values*, onde é passado como parâmetro o valor referente a cada atributo da tabela respeitando o tipo de dados definido na estrutura da mesma.

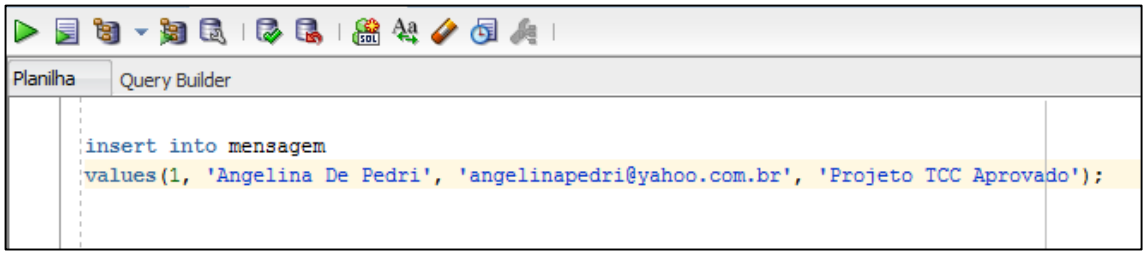

Figura 14: Exemplo de SQL executada no banco de dados relacional para inserção dos dados

Fonte: Autoria própria

4.2. MODELO DE DADOS – PROJETO TCC (NoSQL)

A Figura 15 representa a base de dados Projeto TCC no banco MongoDB NoSQL, onde simula em uma Coleção o relacionamento referente as três tabelas da base de dados SQL. Como o banco de dados NoSQL orientado a documento é livre de estrutura, para esta demonstração foi utilizada a recuperação de dados da tabela CONTABILIDADE através da função *db.contabilidade.find().pretty()*, sem a necessidade de passar um parâmetro, pois compreende somente na estrutura da coleção sem dados.

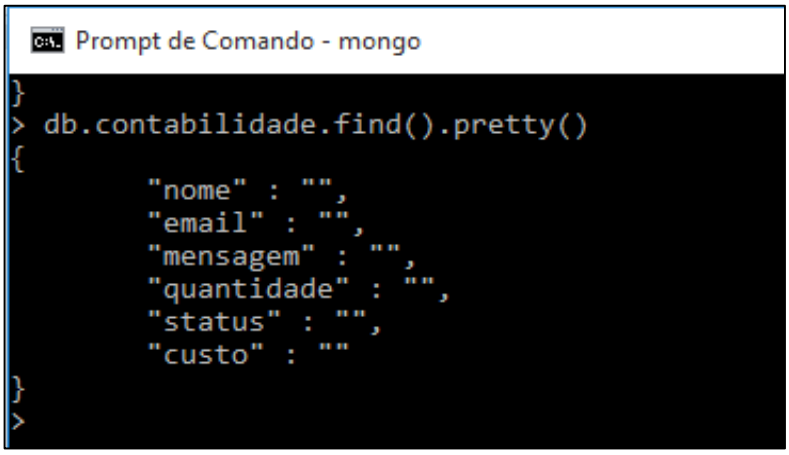

Figura 15: Representação do modelo da base de dados Projeto TCC – NoSQL

Fonte: Autoria própria

4.2.1 Criação da base de dados

Na Figura 16, é representada a criação da base de dados NoSQL utilizando a linguagem BSON, linguagem utilizada no MongoDB. Para criação desta base dados foi utilizado o comando *use* seguido do nome da base de dados projeto TCC.

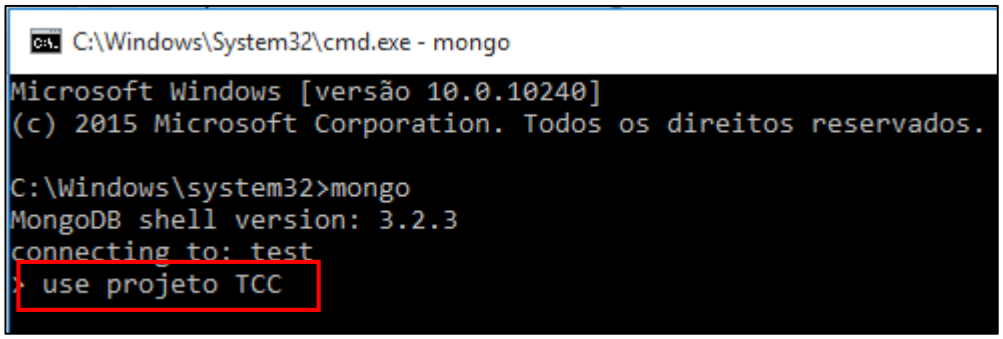

Figura 16: Criação da base de dados Projeto TCC – NoSQL

Fonte: Autoria própria

4.2.2 Criação da Coleção e inserção dos dados

A Figura 17 representa a criação da coleção e a Figura 18 como os dados podem ser inseridos na mesma.

Na Figura 17, a função *db.createCollection()* é utilizada para criação de coleções no NoSQL passando como parâmetro o nome da coleção que será criada.

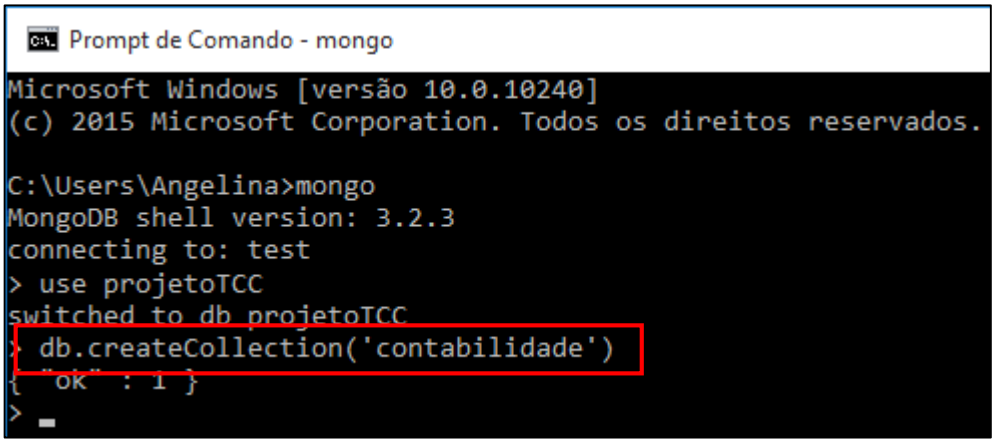

Figura 17: Criação da coleção na base de dados NoSQL

Fonte: Autoria própria

A função *db.contabilidade.insert()* é utilizada para a carga de dados na coleção conforme Figura 18, e os registros são passados como parâmetro, sem ser necessário definir o tipo de dados ou declarar as chaves estrangeiras, já que no NoSQL a chave primaria é criada automaticamente com o nome ID e seu valor atribuído, como auto incremento.

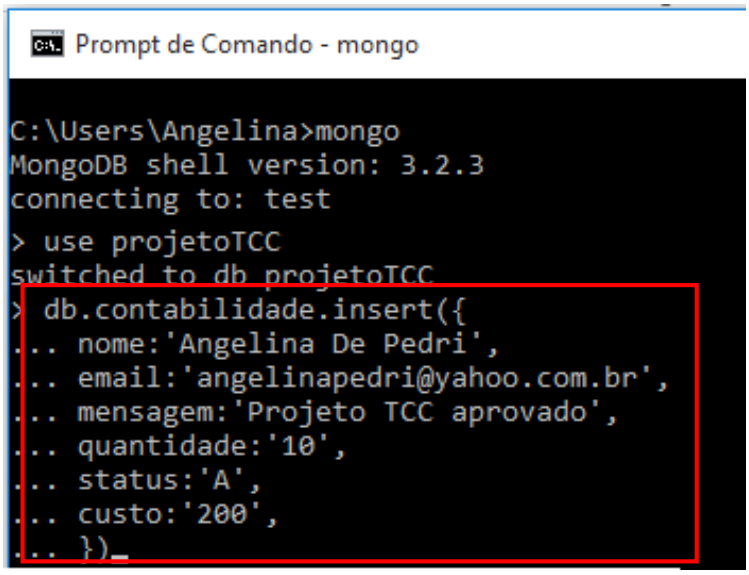

Figura 18: Inserção de dados na coleção – NoSQL

Fonte: Autoria própria

4.3 COMPARAÇÃO ENTRE AS BASES DE DADOS SQL e NoSQL

Neste capítulo serão abordadas as principais diferenças entre um banco de dados SQL – Relacional e um banco de dados NoSQL.

Como exemplificação para esta comparação foi selecionada a recuperação de dados englobando o conceito de agregação seguido das instruções DML *Update* e *Delete* que serão descritos á seguir.

4.3.1 Agregação

As funções agregação são aquelas nativas de consulta utilizadas para agrupar informações, agregando em uma única linha o conteúdo de várias linhas. Dessa forma, os detalhes da informação original são ocultados e as informações são tratadas em conjunto.

Estas são utilizadas quando ocorre a necessidade de relacionamento e união entre determinadas colunas de uma tabela para obter o resultado desejado.

Essas funções agrupam os valores e os retornam em um valor baseado no conjunto de valores dos campos agregados.
A seguir serão apresentas as funções de agregação mais utilizadas referente ao banco de dados SQL e NoSQL e uma exemplificação.

4.3.2 Agregação em um Banco de Dados SQL

Conforme o conceito descrito na subseção anterior sobre agregação será demonstrado na Tabela 1 algumas funções mais utilizadas e na Figura 19 um exemplo de uma instrução DML, *select*, que engloba a utilização da função *SUM*.

Função	**Ação**
COUNT	Retorna o número de linhas
SUM	Retorna a somatória do valor dos campos envolvidos
AVG	Retorna a média aritmética dos campos envolvidos
MIN	Retorna o menor valor da coluna de um grupo de linhas
MAX	Retorna o maior valor da coluna de um grupo de linhas

Tabela 1: Funções de Agregação

Fonte: Autoria própria

Na Figura 19 segue o exemplo da função de agregação *SUM*, na qual é feita a somatória da coluna ‘quantidade’ da tabela CONTABILIDADE, finalizando a função com a utilização da cláusula *group by*, que faz o agrupamento das colunas que foram selecionadas no início da *query* (c.custo_id e cli.nome).

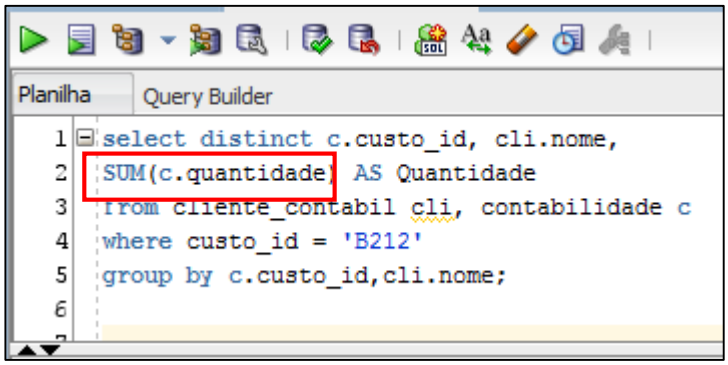

Figura 19: Demonstração da função de agregação *SUM*

Fonte: Autoria própria

4.3.3 Agregação em um Banco de Dados NoSQL

Na Tabela 2 serão apresentadas algumas funções de agregação em um banco de dados NoSQL e na Figura 20 será apresentada a função *MapReduce*, que englobada uma consulta comparada a instrução DML *select*, que está representada na função 'query'.

Função	**Ação**
MAPREDUCE	Processa cada documento e reduz a fase que combina da operação de mapeamento.
SUM	Soma o valor definido a partir de todos os documentos da coleção.
AVG	Calcula a média de todos os valores dados de todos os documentos da coleção.
MIN	Obtém o mínimo dos valores correspondentes de todos os documentos da coleção.
MAX	Obtém o máximo dos valores correspondentes de todos os documentos da coleção.

Tabela 2: Funções e respectivas ações NoSQL

Fonte: Autoria própria

Conforme a Figura 11 a seguir, pode-se observar a instrução *emit* comparada a cláusula *group by,* que está relacionada ao banco de dados SQL, que tem a função de agrupar os atributos selecionados para exibição na *query.*

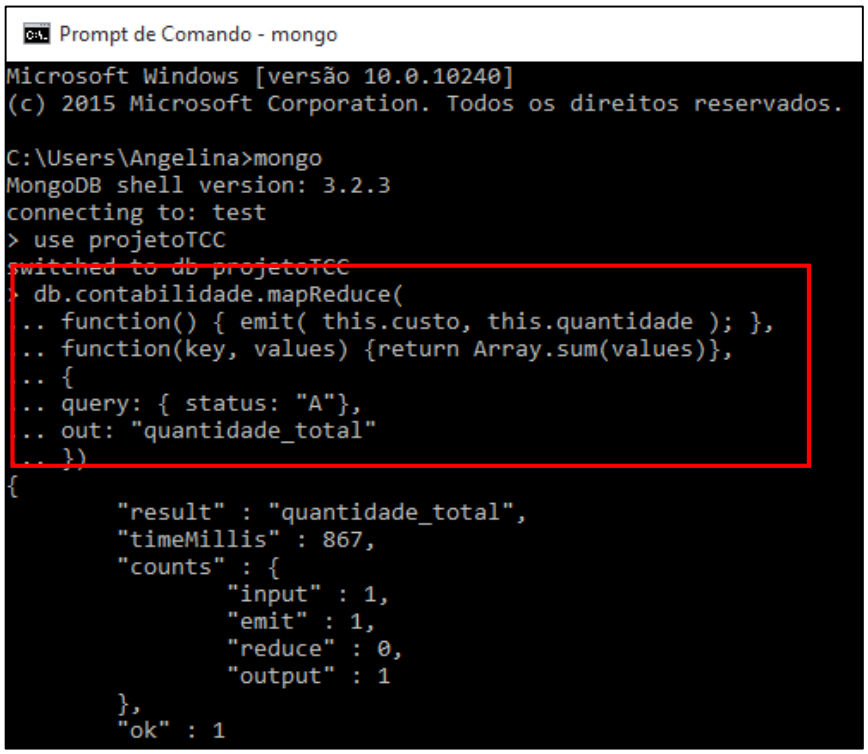

Figura 20: Exemplo de Função de agregação *MapReduce*

4.3.4 Instrução *Update*

A instrução DML *Update* compreende o conceito da manipulação de dados de uma tabela ou coleção, onde os dados existentes na mesma podem ser modificados seguindo a formatação do atributo em questão, diferenciando sua composição em uma base de dados SQL para uma base NoSQL como poderá ser acompanhando nas seções a seguir.

4.3.4.1 Instrução *Update* (SQL)

Segue um exemplo da instrução *Update* na Figura 21 em uma base de dados SQL, onde para esta manipulação de dados foi necessária a utilização da instrução *set*, que aponta o atributo onde o seu conteúdo será modificado e da cláusula *where*, que irá identificar para qual valor de qual atributo selecionado este dado será manipulado.

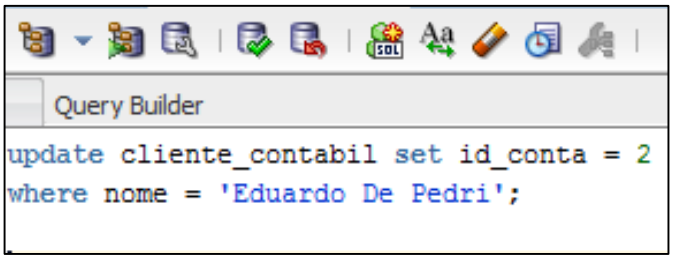

Figura 21: Exemplo da instrução DML *Update* em uma base de dados SQL

Fonte: Autoria própria

4.3.4.2 Instrução *Update* (NoSQL)

A Figura 22 contém uma exemplificação da instrução DML *Update* em uma base de dados NoSQL. Nota-se que a composição para tal não requer a utilização da cláusula *where* para sua execução já que através do método *db.contabilidade.update()* é passado o parâmetro como valor requerido utilizando a instrução *set* para alterar somente a linha que possui registro (chave/valor) desejado. Assim nota-se o modelo BSON para manipulação de dados menos complexo que o modelo SQL.

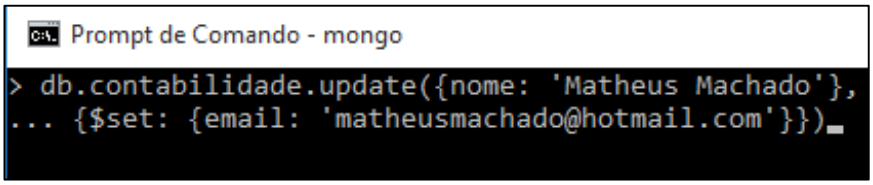

Figura 22: Exemplo da função d*b.contabilidae.update()* em uma base de dados NoSQL

Fonte: Autoria própria

4.3.5 Remoção de registros

Para que seja possível remover registros em uma base de dados SQL é utilizada a instrução DML *Delete*. Esta instrução possibilita a remoção dos dados desejados mantendo a estrutura dos atributos e tuplas, porém, um registro não pode ser deletado se o mesmo possuir uma chave primária em outra tabela que está sendo utilizado como chave estrangeira. Já na base NoSQL é utilizada a função *db.contabilidade.remove()* que equivale ao *Delete* e não possui esta integridade, conforme descrito no Capítulo 2.

4.3.5.1 Instrução *Delete* (SQL)

Na Figura 23 é demonstrada uma *query* composta com a instrução DML *Delete.* É notável a utilização da cláusula *where* e do operador *and* para a localização do registro na tabela e posterior execução do mesmo, já que seria possível neste exemplo que o usuário 'Matheus Machado' pudesse ter enviado mais de uma mensagem, assim teria mais de um registro como o mesmo nome de usuário, sendo necessário filtrar a informação que deseja deletar contida no campo 'mensagem'.

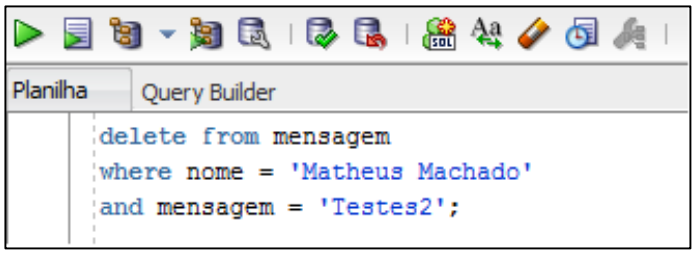

Figura 23: Exemplo da instrução DML *Delete* em uma base de dados SQL

Fonte: Autoria própria

4.3.5.2 Função *db.contabilidade.remove()* (NoSQL)

Para realizar a remoção de dados em uma base NoSQL é utilizada a função *db.contabilidade.remove()*, que equivale a instrução DML *Delete*. Sua composição possui o método e o registro necessário que deverá ser removido, não sendo necessária a utilização de cláusulas. Na Figura 24 a seguir, será demonstrado um exemplo desta função.

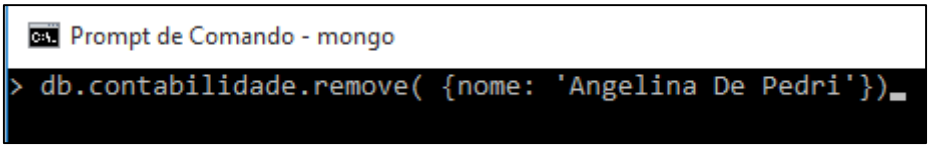

Figura 24: Exemplo da função *db.contabilidade.remove()* em uma base de dados NoSQL

Fonte: Autoria própria

5. CONSIDERAÇÕES FINAIS

Conclui-se com este trabalho que os SGBD's NoSQL são muito importantes para o armazenamento de dados não relacionais a nível de *Big Data* que a cada dia vem crescendo exponencialmente, oferecendo facilidade na consulta dos mesmos.

Este modelo de BD também se ajusta a várias tecnologias de armazenamento e consulta de dados como citado nos capítulos anteriores que auxiliam no retorno das informações com coerência facilitando a tomada de decisões pelas empresas que precisam destes dados de uma maneira mais organizada.

Enquanto o modelo relacional tradicional SQL não perde seu espaço, sendo ainda muito útil para a organização dos dados de sistemas convencionais que utilizam as tabelas, tuplas e atributos para montar seu relacionamento.

No entanto, os benefícios que fazem parte dos BD's NoSQL ainda estão sendo estudados, porém os que já podem ser notados são: a facilidade na elaboração de uma consulta, armazenamento de um grande volume de dados não relacional e menos custoso, no sentido da elaboração das *querys,* do que os bancos de dados SQL.

Esta citação foi confirmada de acordo com o estudo de caso apresentado no Capítulo 4 seguido de suas estruturas, apresentando na prática o modelo orientado a documentos NoSQL MongoDB.

O modelo apresentado possui linguagem prática e estrutura mais simples para elaboração das *querys*, resultando na agilidade das consultas, além de possuir a função MapReduce, que realiza o processamento do documento de uma coleção trazendo a informação estipulada na *query*.

Porém, a área de estudos sobre *Big Data* e banco de dados NoSQL ainda está sendo desmitificada aos poucos, afim de adquirir mais informações e grandes benefícios sobre a mesma, e assim a cada dia novas pesquisas e estudos vão surgindo a fim de contribuir com esta ação.

REFERÊNCIAS BIBLIOGRÁFICAS

Aggregation, Inc 2008-2016. Disponível em: <https://docs.mongodb.com/manual/aggregation/>.
Acessado em: 10 de Junho de 2016.

BARASUOL, R. Érion. **MongoDB uma base de dados orientada a documentos que utiliza orientação a objetos**. 2012. 101 f. Trabalho de Conclusão de Curso - Instituto Municipal de Ensino Superior de Assis, 2012.

BARTH, J. Fabrício. **Uma Introdução á Mineração de Informações na era do *Big Data***. 2012. 75. Tipo de trabalho (Titulação) - VAGASTecnologia e Faculdades BandTec.

BIRGNOLI, T. Juliano, JUNIOR, S. Egon, MIGUEZ, B. Viviane, SANTOS, Neri, SPANHOL, Fernando. **A Intervenção Humana na Qualificação de Processos de Data Mining: Estudo de Caso em uma Base de Dados Hipotética.** Universidade Federal de Santa Catarina.

BRUNO, M. **Introdução ao MongoDB, 2013.** Disponível em: <http://www.pinceladasdaweb.com.br/blog/2013/12/19/introducao-ao-mongodb/>.
Acessado em: 04 de Março de 2016.

CAPETTA, M. Leonardo. **Consulta a banco de dados em linguagem natural**. Omnia Exatas, v.4, n.1, p.72-80, 2011.

Conceitos: Data Warehouse e Data Mining. Disponível em: <http://www.macoratti.net/dwh_dmn.htm>.
Acesso em: 21 de Fevereiro de 2016.

Critérios para Seleção de SGBD NoSQL: o Ponto de Vista de Especialistas com base na Literatura. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2014/0012.pdf>.
Acesso em: 18 de Fevereiro 2016.

Data Mining x Data Warehouse. Disponível em: <https://sites.google.com/site/mineracaodedados1b/data-mining-x-datawarehouse>.
Acesso em: 22 de Fevereiro de 2016.

DOURADO, Joana. **Semantix – Treinamentos.** Disponível em: <http:// www.semantix.com.br>.
Acesso em: 05 Dezembro de 2013.

GOLDMAN, A., KON, F., JUNIOR, F. P., POLATO, I, PEREIRA, R. F.**. Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades**.

GOMES, M. Heitor., HAUTH, G. Luiz., CARVALHO, R. Deborah. **Mineração de dados temporal: Descoberta de Regras de Causa e Efeito**. Faculdade de Ciências Exatas e Tecnologia (FACET) – Universidade Tuiuti do Paraná – Curitiba – PR – Brasil.

I**ntrodução ao MongoDB: um banco de dados NoSQL**. Disponível em: <http://www.itexto.net/devkico/?p=682>.
Acesso em: 18 de Fevereiro de 2016.

JUAN, Pablo. **Configurando ambiente MongoDB no Windows**. Disponível em: <https://pablojuancruz.wordpress.com/2014/09/03/configurando-ambiente-mongodb-no-windows/>.
Acesso em: 22 de Janeiro de 2016.

KIMBALL, R. **The Data Warehouse Toolkit**: Guia Completo para modelagem dimensional. Tradução da segunda edição. Rio de Janeiro: Campus Ltda, 2002.

KOHONEN, Teuvo**. An Introduction to Neural Computing**. Finland: Helsinki University of Tecnology, pp. 3-16, 1988.

KOHONEN, Teuvo**. Self-Organization and Associative Memory**. 2ª Edição. USA: Spring-Verlag, 1989.

KOHONEN, Teuvo. **The self-organizing map**. Proceedings of the Institute of Electrical and Electronics Engineers, vol.78, pp. 1464-1480, 1990.

LENNON, Joe. **Explore o MongoDB, 2011**. Disponível em: <https://www.ibm.com/developerworks/br/library/os-mongodb4/#ibm-pcon>.
Acesso em: 29 de Fevereiro de 2016.

LIPPMANN, Richard. **An Introduction of Computing with Neural Nets**. IEEE Computer Society, v.3, nº 4, pp. 4-22, abr.1987.

LÓSCIO, F. Bernadette; OLIVEIRA, R. Helio; PONTES, S. C. Jonas. **NoSQL no desenvolvimento de aplicações Web colaborativas**. In: VIII Simpósio Brasileiro de Sistemas Colaborativos, Brasil, 2011. Porém, o artigo foi publicado em 2014. Disponível em: <http://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf>.
Acesso em: 18 de Fevereiro de 2016.

MAYER-SCHÖNBERGER, Viktor., CUKIER, Kenneth. ***BIG DATA*, Como Extrair Volume, Variedade, Velocidade e Valor da Avalanche Cotidiana**. Editora Elsevier Ltda, 2013.

MEDEIROS, Higor. **Introdução ao MongoDB.** In Universidade do Vale do Rio dos Sinos em Ciência da Computação**.** Disponível em: <http://www.devmedia.com.br/introducao-ao-mongodb/30792>.
Acesso em: 23 de Fevereiro de 2016.

MORAIS, S. Alexandre; PRADO, P. V. Edmir. **CRITÉRIOS DE SELEÇÃO DE SGBD NOSQL EM ORGANIZAÇÕES BRASILEIRA**. Publicado na Revista Eletrônica de Sistemas de informação ISSN 1677-3071, no ano de 2014. Disponível em: <http://www.periodicosibepes.org.br/ojs/index.php/reinfo/article/view/1927>.
Acesso em: 18 de Fevereiro de 2016.

NASCIMENTO, O., J., Rafael. **Mineração e Análise de Dados em SQL**. Disponível em: <http://www.devmedia.com.br/mineracao-e-analise-de-dados-em-sql/29337>.
Acesso em: 04 de Agosto de 2014

NASCIMENTO, Jean. **3 razões para usar MongoDB.** Disponível em: <http://imasters.com.br/artigo/18334/mongodb/3-razoes-para-usar-mongodb>.
Acesso em: 29 de Fevereiro de 2016.

PEREIRA, W., HEINRICH, T., SCHROEDER, R. . **Avaliação de Desempenho de Sistemas Relacionais para Armazenamento de dados RDF.**

PINA, Jeronimo. **Hearing from the Thought Leaders**. Publicado no ano de 2015 pela IDC Brasil. Disponível em:<http://br.idclatin.com/newsletters/laupdates/pt/section1_15_may_2.aspx>.
Acesso em: 21 de Fevereiro de 2016.

TAURION, Cezar. **Coletânea de posts publicados no Blog developerWorks em2012developerWorks Brasil**. Disponível em: <http://www.ibm.com/developerworks/blogs/page/ctaurion >.
Acesso em: 07 Outubro de 2013.

TURNER, Vernon. **The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things.** Disponivel em: <http://www.emc.com/leadership/digital-universe/2014iview/index.htm>.
Acesso em: 12 de Fevereiro de 2016.

REIS**,** Thiago. **DATA WAREHOUSE E DATA MINING.** Disponível em: <http://tecnologiae-e-negocios-thiagoreis.blogspot.com.br/2010/04/data-warehouse-e-data-mining.html>.
Acesso em: 22 de Fevereiro de 2016.

ROLIM, B. V.; SILVA, R. M., SCHMELZER, V., BRAZ, F. J., SILVA, E. **Estratégias para importação de grandes volumes de dados para um servidor PostgreSQL.**

SBPJor – Associação Brasileira de Pesquisadores em Jornalismo, IX, 2011, Rio de Janeiro. **Jornalismo Computacional em função da Era do *Big Data***: 2011.12.
Significado de Data Mining. Disponível em: < http://www.significados.com.br/data-mining/>.

Acesso em: 22 de Fevereiro de 2016.

STEPPAT, Nico. **Bancos de dados não relacionais e o movimento NoSQL. Atrigo publicado no ano de 2009.** Disponível em: <http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-nosql/>.
Acesso em: 18 de Fevereiro de 2016.