Introdução

Devido ao fato de estarmos inseridos em um mundo globalizado onde a informação é o principal item a ser comercializado para diversos fins, surgiu a necessidade de armazenar grandes volumes de dados para que estes fossem tratados e comercializados em forma de informação. Neste contexto foram criadas tecnologias capazes de suprir esta demanda e também de aperfeiçoá-la, tornando-a mais eficiente e acessível a diversas finalidades, desde armazenar produtos em estoque de uma simples empresa até o armazenamento do histórico de comportamento de bilhões usuários.

Desde sua criação, o banco de dados evoluiu para atender diferentes demandas. O que antes armazenava apenas dados locais, hoje conectado à rede mundial de computadores (WLAN), armazena dados de inúmeros dispositivos espalhados pelos mais diversos lugares e também permite estabelecer a conexão de diferentes sistemas para acessá-los remotamente.

Com o intuito de construir sistemas de gerenciamento de banco de dados (SGBDs) que fossem menos abstratos e mais eficientes, empresas do ramo de tecnologia começaram a trabalhar no desenvolvimento de novos conceitos, o caso de maior implementação foi o Diagrama Entidade Relacionamento (DER), o qual pode-se dizer que se tornou um padrão para modelagem de dados em forma de tabelas. Aliado a isto, houve uma padronização na linguagem de consulta e manipulação de dados do banco, padronizando assim a linguagem SQL (*Structured Query Language*).

Com o passar do tempo, foi visto que criar tabelas demorava muito e armazenava informações repetidas e/ou desnecessárias, então uma outra forma de armazenamento de dados emergiu, o NoSQL (*Not-only Structured Query Language*). Como uma de suas principais características, a maleabilidade da "tabela", os dados são centralizados, removendo assim a necessidade do referenciamento de chaves (*primary key* e *foreign key*). Sendo assim, o banco de dados fica mais leve, e consequentemente mais rápido, o que facilita no tratamento de grandes volumes de dados (big data).

Quando nos referimos aos bancos de dados open source que são relacionais, há diversas opções disponíveis na internet, entre os mais famosos temos o MySQL, PostgreSQL e MariaDB. No caso dos bancos de dados não relacionais, sobressaem-se o MongoDB, Redis e Cassandra.

Para objeto de estudo neste trabalho escolhemos o MySQL e MongoDB como banco de dados *open sources* e o SQL Server como banco de dados *closed source*. O objetivo deste trabalho consiste em comparar diferentes características de cada um e apresentar um *framework* capaz de realizar a conversão do SQL para NoSQL.

Desenvolvimento

Definição

Os bancos de dados não relacionais são classificados por chave-valor, banco de dados de documentos, armazenamento em famílias de uma coluna e banco de dados de grafos diferente do banco de dados relacional representa o banco de dados como uma coleção de relações.

Referência: RABELO, Daniel Ferreira; CÂNDIDO, Marco Vinicius Isecke. ANÁLISE DE DESEMPENHO DE BANCO DE DADOS NOSQL EM UM SISTEMA QUE UTILIZA UM BANCO DE DADOS RELACIONAL E NÃO RELACIONAL PARA ARMAZENAMENTO DE DADOS. 2017.

Já NoSQL é uma metodologia que usa chave primaria, mas guarda todos os dados em um único local como se fosse uma pilha de dados com várias pranchetas. Embora o banco de dados relacional predomine há algumas décadas, os conceitos relacionados ao modelo não-relacional – o NoSQL – estão sendo cada vez mais difundidos. Entretanto, existem vantagens e desvantagens em relação a sua adoção, mas em se tratando de ambientes Big Data, a sua aplicabilidade é de extrema relevância, pois possibilita que os dados sejam processados, armazenados e manipulados com maior desempenho, pelo fato deste modelo ser projetado para tal ambiente. Assim, podemos dizer que o NoSQL surge como uma alternativa aos bancos de dados convencionais, suprindo as restrições (desempenho) ao tratar grandes volumes de dados (Big Data).

Referência: PADILHA, Renata Junges; SCHUCH, Regis Rodolfo; DE LIMA, Claudinei. NoSQL EM CONTEXTOS DE BIG DATA. Revista GEDECON-Gestão e Desenvolvimento em Contexto, v. 5, n. 1, p. 148-151, 2017.

Contexto

Devido ao sucesso na construção de *softwares* de gerenciamento de banco de dados, a demanda por este tipo de ferramenta foi crescendo e a implementação de um *software* proprietário começava a esbarrar nas propriedades de patentes, o que tornava inviável realizar alterações do comportamento do *software* para atender casos específicos. Além disso, nem sempre todos os termos de um contrato de licença satisfaziam o contratante, o qual era obrigado a submeter-se devido à falta de alternativas ou então se abster de tal ferramenta.

//INSERIR O CONTEXTO DO SQL SERVER

Neste cenário surgiu duas vertentes, ambas já existentes no *software* em geral, mas que passou então a ser adotado para o banco de dados. A política de código fechado (*closed source*) e a política de código aberto (*open source*). Para diferenciar cada vertente, podemos definir *open source* como um software que os usuários podem usar, copiar, compartilhar, estudar, mudar e melhorar; isto inclui realização de mudanças no código fonte. Ou seja, o usuário (tanto individual como corporativo) pode fazer uso do *software* *open source* como bem entender. Já um *software* de código fechado, somente a empresa que é portadora dos direitos autorais do software tem o poder de alterar/visualizar o código-fonte de seu *software*.

Os bancos de dados *open source* tem a vantagem de serem mais flexíveis e inovações podem ser feitas mais rapidamente do que nos bancos de dados de código fechado. Isso se deve ao fato de que possuem um suporte e milhares de programadores independentes que estão constantemente testando e corrigindo erros. Outra vantagem do banco de dados *open source* em relação ao suporte que possui da comunidade é que ele não necessariamente é dependente da empresa/pessoa que originalmente o desenvolveu.

Atualmente SGBD mais popular dentro do contexto de banco de dados *open source* e que possui a linguagem SQL é o MySQL, o qual tem o suporte da Oracle Corporation. Foi primeiramente desenvolvido por David Axmark e Michel Monty Widenius por volta de 1995. Segundo eles, o MySQL surgiu por causa da necessidade de ter um banco de dados mais flexível e rápido que o banco de dados mSQ - banco de dados que era usado por eles na época.

O GPL (GNU General Public License) é uma licença de softwares gratuitos utilizada pelo MySQL. A GPL possibilita aquela liberdade para os usuários de rodar, estudar, compartilhar e modificar o software segundo seus próprios interesses.

//INSERIR O CONTEXTO DO MONGODB AQUI

O NOSQLayer É um framework capaz de realizar automaticamente a migração de dados entre banco de dados relacionais e NoSQL, mantendo a semântica do banco original. Há dois módulos no NoSQLayer: módulo de migração de dados e o módulo de mapeamento de operações SQL O NoSQLayer também possui um módulo intermediador que realiza a tradução da operação, recupera os dados e os coloca no formato esperado e retorna para a aplicação de forma totalmente transparente, ou seja, intercepta as operações SQL enviadas pela aplicação. Atualmente só disponível de MySql para MongoDB.

Referência: link do começo sobre NoSQLayer

Comparação entre bancos de dados

**Tabela comparativa entre bancos de dados**

|  |  |  |  |
| --- | --- | --- | --- |
|  | MongoDB | MySQL | SQL Developer |
| Modelo | NoSQL | SQL | SQL |
| Política | Open source | Open source | Open source |
| Contextualização | Big data |  |  |
| CRUD | Anexo X.1 | Anexo X.2 | Anexo X.3 |
| Suporte | Comunidade  e  tercerizado | Comunidade  e  tercerizado | Official |
| Necessidade de normalização 3NF | Não | Sim | Sim |

**Tabela comparativa entre performance no banco de dados**

|  |  |  |  |
| --- | --- | --- | --- |
|  | MongoDB | MySQL | SQL Developer |
| Tamanho do arquivo | ↑ | ↑↑ | ↑↑ |
| Alteração | \* | ↑↑↑ | ↑↑↑ |
| Consulta | \* | ↑↑↑ | ↑↑↑ |
| Inserção | \* | ↑↑↑ | ↑↑↑ |
| Remoção | \* | ↑↑↑ | ↑↑↑ |

Legenda: Extremamente alto: \*; Alto: ↑↑↑; Médio: ↑↑; Baixo: ↑

//Comparação – texto corrido.

//Performance – texto corrido.

//Sugestão de aplicação – MySQL, Mongo e SQL Server.

CONCLUSÃO

Anexo

//CRUD – MySQL, MongoDB