Universidade Federal do ABC
Bacharelado em Ciência e Tecnologia

Matheus Ribeiro Barison Martins Silva

SOLUÇÕES DE BANCO DE DADOS

Ánalise comparativa entre tipos de banco de dados

Projeto de pesquisa apresentado como exigência parcial para avaliação quadrimestral da disciplina Projeto Dirigido do Curso de Bacharelado em Ciência e Tecnologia da Universidade Federal do ABC.

Santo André
2021

SUMÁRIO

RESUMO

Banco de dados é uma ferramenta imprescindível no contexto atual da nossa sociedade, visto que, produzimos muitos dados em simples ações cotidianas, como o mero ato de visualizar o *feed* na página de alguma rede social. Todos esses rastros de informações que deixamos para trás podem ser convertidos em um mapa comportamental que auxilia empresas a melhorar seus serviços.
É exatamente nesse cenário que o banco de dados mostra seu valor e, como se trata de um mecanismo muito importante, a evolução de seus modelos e abordagens é algo inevitável e que acontece todos os dias. O NewSQL é uma classe moderna de sistema gerenciador de banco de dados (SGBD) que tentar suprir algumas carências que os modelos anteriores não conseguiam.
Este projeto visa propocionar discussão teórica e comparativa acerca do modelo de banco de dados mais tradicional (tipo relacional) e os emergentes (tipo não relacional e NewSQL), além de instigar estudos relacionados as novas arquiteturas de SGBD. A discussão comparativa será embasada em uma análise de desempenho entre os três modelos de banco de dados apresentados, com o auxílio de benchmarks compatíveis com os modelos de banco de dados que serão apresentados.

Palavras-chave: banco de dados; SQL; NoSQL; NewSQL; BigData; SGBD.

1. INTRODUÇÃO E JUSTIFICATIVA

Nossa espécie vem deixando registros de sua existência há pelo menos 45000 anos, como sugere um artigo da revista Science, com autoria de (Brumm et al., 2021) retratando uma antiga pintura rupestre, na caverna de Leang Tedongnge, localizada na Indonésia. A quantidade de informações que nós estamos gerando ao longo de nossa história está crescendo exponencialmente, nossos avanços nas áreas de tecnologias Web e a disseminação de dispositivos capazes de se comunicar virtualmente pela sociedade originam uma quantidade exorbitante de dados pela rede a todo o momento, conhecidos como *Big Data*.

O modelo de banco de dados mais utilizado hoje em dia ainda é o modelo baseado em entidades e relacionamentos, que foi criado em 1976 por Peter Chen. O SQL (Structure Quey Language) é uma linguagem de computador que é usada para interagir com o banco de dados, esse modelo é conhecido como banco de dados relacional. Quando é necessário recuperar um dado do banco, é utilizada a linguagem SQL para fazer a requisição, na qual será processada pelo SGBD (Sistema de Gerenciamento de Banco de Dados) que retorna a informação requisitada (GROFF & WEINBERG, 1999).

Todavia, a abordagem SQL foi concebida num contexto tecnológico em que o Big Data ainda não era claro e, portanto, não se imaginava a quantidade absurda de dados que seria necessário processar nos modelos mais recentes. Motivados por esses desafios, surgiram novos tratamentos para os bancos de dados conhecidos como NoSQL (Not Only SQL) que proporcionam melhores propriedades no contexto de grande quantidade de informações, contribuindo com uma alta disponibilidade e escabilidade. Muitos aplicativos, no entanto, são incapazes de usar esses sistemas NoSQL, pois, eles dependem de fortes requisitos transacionais e de consistência, conhecidos como propriedades ACID que caracterizam transações OLTP (PAVLO & ASLETT, 2016).

É nesse contexto que surge o NewSQL, que são classes modernas de SGBDs que buscam fornecer o mesmo desempenho do NoSQL para transações OLTP e, por outro lado, ainda oferecer as importantes propriedades ACID para transações que o SQL proporciona.

Este projeto busca fomentar as discussões relacionadas a esse novo modelo de banco de dados conhecido como NewSQL, visto que o material que circunda o tema

ainda está bastante escasso nos repositórios acadêmicos e, além disso, propor uma comparação entre os três modelos de banco de dados apresentados e mensurar o desempenho de cada um frente a um apanhado de dados específicos.

Com isso, proporcionar um embasamento teórico que proporcionará a implementação de testes empíricos em bancos de dados, mapeando as melhores arquiteturas frente a uma massa de dados específicas, a fim de auxiliar na escolha de desenvolvedores *web* na hora de decidir qual modelo de banco de dados usar para compor a sua aplicação.

2. OBJETIVOS

O principal objetivo deste projeto é mapear e comparar três soluções de bancos de dados diferentes, cada uma baseada em um dos três principais modelos disponíveis no mercado hoje em dia, sendo eles o tipo relacional (*SQL*), não relacional (*NoSQL*) e o mais recente que abrange conceito dos dois anteriores, o *NewSQL*. Portanto, este estudo deverá seguir os seguintes objetivos específicos:

- Revisar a bibliografia;
- Definir três bancos de dados que seguem os pressupostos;
- Definir os *benchmarks* adequados aos modelos escolhidos;
- Preparar os materiais *(hardwares);*
- Instalar os programas *(softwares);*
- Realizar os testes;
- Comparar os dados e analisar os resultados.

3. DESENVOLVIMENTO

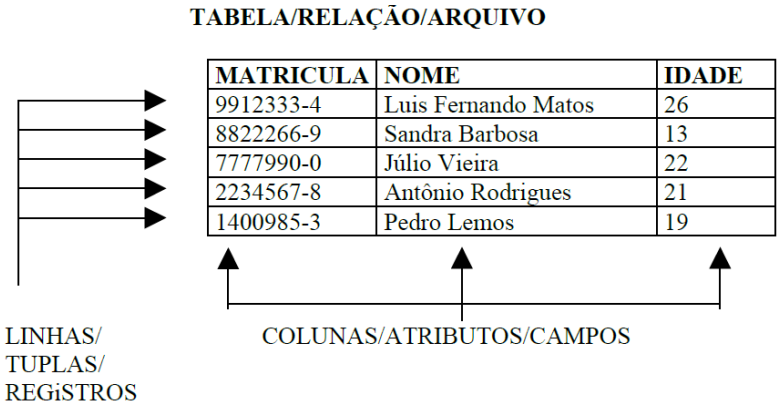
a. Fundamentação teórica

Banco de dados relacional

Um banco de dados relacional é um banco de dados onde todos dados visíveis pelo usuário são estritamente organizados em tabelas de valores de dados, e ainda refletir as operações que serão feitas no banco de dados nas tabelas usadas (GROFF & WEINBERG, 1999).

Portanto, no modelo relacional a principal construção para a representação de dados é a relação entre tabelas, cada relação é composta por tuplas (registros) e atributos (colunas), onde cada registro na tabela é identificado por um campo chave que contém um valor único. As tabelas se relacionam a partir de identificadores (OLIVEIRA et al., 2018).

Figura 1 – Tabela relacional



Fonte: Marcel mesmo, 2011.

Como afirma Oliveira et al. (2018) no que tange os relacionamentos entre tabelas e na própria identificação de certa coluna, as chaves (identificadores) são de suma importância e existem dois tipos:

1) Chave primária: (PK – Primary Key) é o identificador de cada registro, que justamente serve para dar unicidade a uma tupla, portanto jamais deve se repetir.
2) Chave estrangeira: (FK – Foreign Key) é a chave formada por intermédio de um relacionamento com alguma chave primária de outra tabela.

Linguagem SQL

É a linguagem padrão usada para manipular bancos de dados do tipo relacional e consiste em aproximadamente quarenta comandos, onde cada um faz uma

requisição específica para o SGBD, tal como criar uma nova tabela, recuperar certo dado ou até inserir novos dados no banco (GROFF & WEINBERG, 1999).

A linguagem SQL pode ser entendida como uma divisão entre quatro grupos principais de sintaxes, sendo eles:

Data Definition: Comandos que definem estruturas de dados, os principais comandos são:

CREATE TABLE: Adiciona uma nova tabela à base de dados.

ALTER TABLE: Muda a estrutura de uma tabela existente.

DROP TABLE: Remove uma tabela do banco de dados.

Data Manipulation: Úteis para manipular dados que já estão inseridos na base de dados, os principais comandos são:

SELECT: Recupera dados do banco de dados.

INSERT: Adiciona novas colunas de dados ao banco de dados.

DELETE: Remove colunas de dados do banco de dados.

UPDATE: Atualiza dados já existentes no banco de dados.

***Access Control*:** Como o nome indica, esse subconjunto de comandos serve para garantir a segurança interna do banco de dados, com eles o administrador pode definir aspectos de autorização de dados e licenças de usuários. Possui dois comandos:

GRANT: Concede permissões aos usuários

REVOKE: Revoga as permissões concedidas pelo comando anterior.

Transaction Control: Possui três comandos que dizem respeito ao controle de transações no banco de dados, sendo eles:

SET TRANSACTION: Define as características de acesso aos dados da transação atual.

COMMIT: Efetiva as modificações provocadas pela transaçao atual.

ROLLBACK: Aborta as modificações provocadas pela transação atual.

Microsoft SQL SERVER

Trata-se de um gerenciador de banco de dados relacional que utiliza a linguagem SQL para acessar e fazer as manipulações dos dados que se encontram no banco. A camada de protocolo introduz o que é requisitado para o SQL Server, todas as operações que podem ser chamadas para o servidor ocorrem pelo TDS

(Tabular Data Stream). O TDS é um protocolo da camada de aplicação que tem como função fazer a comunicação entre o cliente e o servidor (OLIVEIRA et al., 2018).

NoSQL

Os bancos de dados NoSQL surgiram a partir de uma lacuna que foi deixada para trás, portanto o NoSQL não tem como missão substituir os bancos de dados relacionais, mas sim propor algumas soluções, na qual, em cenários específicos sua implementação pode ser mais vantajosa.

Strauch (2016), lista algumas vantagens que a tecnologia NoSQL pode proporcionar, sendo os principais recursos:

- **Evitar complexidade desnecessária:** Bancos de dados relacionais proporcionam uma variedade de recursos para manter a integridade dos dados, todavia, algumas aplicações não exigem todo esse repertório de recursos.
- **Alta taxa de transferência:** Bancos de dados NoSQL podem fornecer uma taxa de transferência de dados significamente maior do que os SGDB's tradicionais, o artigo cita o exemplo da aplicação do Google, *MapReduce* que é capaz de processar 20 pentabytes por dia armazenados no *Bigtable.*
- **Escalabilidade horizontal:** A escalabilidade basicamente é a capacidade de atender ao aumento das necessidades que determinado serviço demanda.

Em constraste com o banco de dados relacional, os bancos de dados NoSQL são projetados para proporcionar uma escalabilidade horizontal muito eficiente, tudo isso sem necessitar de incrementos na parte de *hardwares.* Ellis (2009, apud STRAUCH, 2016) argumenta que é facil escalar operações de leitura por replicação dos dados e distribuição das cargas nessas replicas. Além disso, uma das grandes vantagens da arquitetura NoSQL é que sua lógica pode ser implementada por qualquer linguagem de programação, evitando APIs complexas que geralmente um servidor SQL utiliza.

Key-/ Value-Stores

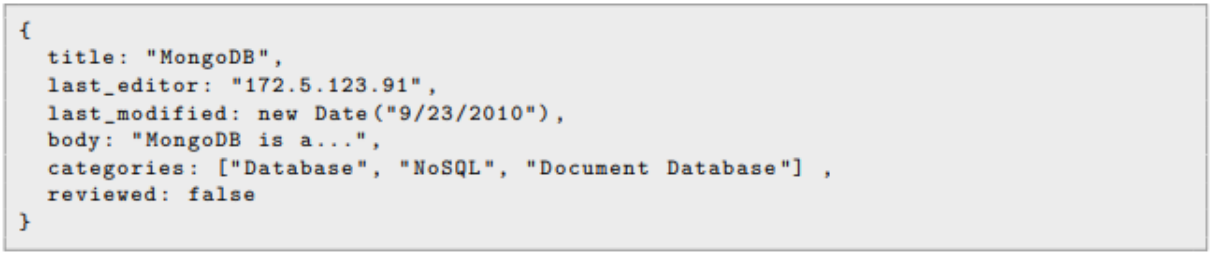
Ambos os modelos possuem um modelo de dados em comum, um mapa que permite ao cliente fazer a requisição dos dados por chaves. Basicamente os objetos do banco de dados são indexados por chaves, que possibilitam sua busca por elas.

(OLIVEIRA et al., 2018). Alguns bancos de dados que usam esse padrão são: Amazon's Dynamo, Couchbase, Tokyo Cabinet, Azure Table Storage, entre outros.

Banco de dados por documentos

Outra classe do NoSQL, são os bancos de dados orientados por documentos. Um documento consiste em campos nomeados que possuem uma chave/nome e seu respectivo valor. Cada campo deve ser único dentro do documento a qual ele pertence e seu valor pode ser um conjunto de caracteres (*string*), um número inteiro, uma data, um valor booleano (verdadeiro ou falso) (STRAUCH, 2016).

Figura 2 – Documento em JSON



Fonte: NoSQL Databases, 2016.

A imagem acima representa um tipo de documento em formato de JSON, além desse formato existem os documentos em formato de XML, que seguem uma lógica parecida ao do JSON e também são amplamente utilizados, além de outros tipos de documentos.

MongoDB

É um sistema gerenciador de banco de dados (SGBD) orientado a documentos, com um esquema livre e é escrito em liguagem C++. De acordo com seus desenvolvedores, o principal objetivo desse SGBD é preencher a lucuna entre o rápido e altamente escalável sistema *key-/ values-stores* e as tradicionais propriedades dos bancos relacionais. A abstração e a unidade de dados armazenáveis no MongoDB é orientado a documentos. O MongoDB utiliza BSON, bastante similar ao JSON, entretanto utiliza seus valores no sitema binário por questões de eficiência (STRAUCH, 2016).

Como aponta Oliveira et al. (2018), algumas características importantes desse SGBD são:

- **Queys Ad hoc:** É uma consulta de uso específico com o retorno definido.
- **Índice:** Qualquer campo do documento que guarda o dado pode ser

indexado, aumentando a eficiência da consulta.

- **Balanceamento de carga:** O escalamento horizontal pode ser distribuído entre os servidores, onde cada servidor recebe o mesmo dado permitindo a identificação de duplicidades.
- **JavaScript:** O MongoDB permite o uso da linguagem JavaScript para execução de códigos diretamente no servidor.

NewSQL

Banco de dados NewSQL é uma nova tecnologia que busca unir o melhor dos dois mundos dos outros dois modelos. Essa classe busca oferecer o mesmo desempenho de escabilidade do NoSQL ao mesmo passo que busca manter as importantes propriedades ACID dos bancos de dados relacionais (KNOB et al., 2019).

Essa classe de banco de dados compartilha algumas características gerais, como (i) Ter um controle de concorrência de esquema *lock-free;* e, (ii) Possuir uma arquitetura distribuída *shared-nothing* (STONEBRAKER, 2012). Todavia, cada sistema NewSQL possui suas peculiaridades, tendo características próprias de como executar as operações de manipulação de dados (KNOB et al., 2019).

NuoDB

O NuoDB é um banco de dados que implementa a solução em NewSQL e é baseado no conceito de *cluster,* onde um *cluster* é um conjunto de nodos de arquivos de transações. No conceito de *cluster*, os computadores são denominados como nodos, onde cada computador pode fazer parte de vários *clusters,* e basicamente esses nodos são usados para fazer o processamento de transações SQL (CINCURA, 2012).

A arquitetura desse banco é constituída por duas camadas, *TE (Transaction processing)* e SM (*Store Management*). TE é a camada que é constituída pelos nodos citados acima, e é nessa camada que as requisições SQL são feitas. Quando qualquer aplicação faz requisições ao banco esta camada cria *caches* em memória para a carga de trabalho requisitada. Já a camada SM é um nodo de processamento que possui componentes em memória e em disco rígido (KNOB et al., 2019).

b. Metodologia

Para conduzir os testes de desempenho e comparar os três modelos de arquitetura apresentados na seção anterior, usaremos os três tipos de bancos de dados relatados, um para cada arquitetura, sendo eles: *Microsoft SQL Server* (banco de dados relacional); MongoDB (banco de dados não relacional) e o NuoDB (NewSQL). Além do uso dos benchmarks, HammerDB (compatível como o Microsoft SQL Server) e o YCSB (compatível com o MongoDB e o NuoDB).

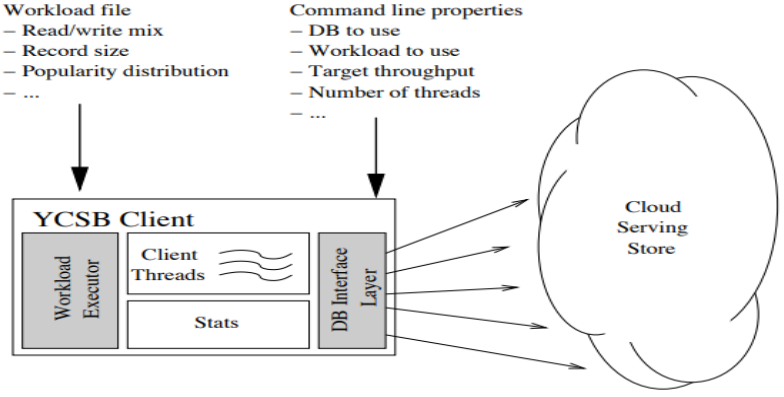
Benchmark

O conceito de *Benchmarking* diz respeito a análises estratégicas das melhores práticas que empresas de um mesmo ramo estão empregando no mercado. '*Benchmark'* significa 'referência', e trata-se de uma ferramenta de gestão que tem como finalidade aprimorar processos, produtos e serviços (CASTRO, 2020). No ramo da computação, o *benchmark* possui o mesmo princípio, servindo como uma ferramenta que nos permite comparar o desempenho entre diferentes dispositivos, em geral o benchmark é aplicado na avaliação de hardwares, todavia, também pode ser empregado para avaliação de softwares, como será o caso deste projeto.

YCSB

O YCSB (Yahoo! Cloud Serving Benckmark) cliente é programa escrito em Java que tem como finalidade gerar dados que serão carregados em uma base de dados, além de forçar operações que demandam cargas de trabalho. Sua arquitetura está representada na figura 3.

Figura 3 – Arquitetura cliente YCSB



Fonte: Benchmarking Cloud Serving Systems with YCSB, 2010.

O processo básico desse *benchmark* é que quem executa as cargas de trabalho simula os vários caminhos que um cliente faria, na qual, são chamados de threads. Cada thread conduz uma série de operações sequenciais que fazem requisições para a camada de interface do banco de dados, quem servem tanto para carregar o banco de dados quanto para executar a carga de trabalho sobre ele (COOPER, et al., 2010).

O YCSB gera carga de trabalho e um pacote com carga padrão que é muito útil para avaliar o desempenho de hardwares e softwares, através de cargas de leitura e escrita intensa, consulta completa em tabelas, dentre outras funções. A execução de um teste de desempenho realiza diversas escolhas aleatórias que serão traduzidas em requisições a interface do banco, tudo isso regido por distribuições randômicas (KNOB et al., 2019).

HammerDB

Assim como o YCSB, o HammerDB é uma ferramenta geradora de cargas de trabalho para bancos de dados. É utilizado para criar esquemas de testes, carrega-los com dados e simular diversas requisições de clientes virtuais a base de dados com a finalidade de explorar cenários que podem ser deficientes em softwares ou em hardwares. Projetado específicamente para ser usado em bancos de dados relacionais, diferente do YCSB.

4. CRONOGRAMA DE ATIVIDADES

O presente projeto tem uma previsão de ser realizado em um semestre, seguindo o cronograma da tabela 1.

Tabela 1 – Cronograma do projeto

Fases	1º mês	2º mês	3º mês	4º mês	5º mês	6º mês
Aprovação do projeto	x					
Revisão bibliográfica	x	x	x			
Preparação do material			x	x		
Realização dos testes				x		
Análise dos resultados				x	x	
Preparação do relatório					x	x

5. REFERÊNCIAS

BRUMM, Adam; OKTAVIANA, Adhi; BURHAN, Basran; HAKIM, Budianto; LEBE, Rustan; ZHAO, Jian; SULISTYARTO, Priyatno; RIRIMASSE, Marlon; ADHITYATAMA, Shinatria; SUMANTRI, Iwan; AUBERT, Maxime. Science Advances. Revista Galileu, volume 7, número 3, Janeiro, 2021.

CASTRO, Ivan. O que é benckmarking e qual a sua importância para o Marketing Digital. Rock content, junho 2020. <Benchmarking: o que é, como fazer e qual a sua importância (rockcontent.com)>. Acesso em: 03/12/2021.

CINCIRA, Jiri. NuoDB – starting with “NewSQL” database. Tabs over spaces, 2012. Disponível em: <https://www.tabsoverspaces.com/232827-nuodb-starting-with-newsql-database>. Acesso em: 29/11/2021.

COOPER, Brian; SILBERSTEIN; Adam, TAM; Erwin, RAMAKRISHNAN; Raghu, SEARS, Russell. Benchmarking cloud serving systems with YCSB. *In:* ACM Symposium on Cloud Computing (SoCC), Indianapolis, Indiana, June 2010.

FREITAS, Marcel. Sistema de banco de dados relacional. Marcel mesmo, 2011. Disponível em: <Marcel mesmo: Sistema de banco de dados relacional>. Acesso em: 23/11/2021.

GROFF, James; WEINBERG, Paul. SQL: The complete reference. 1ª edição. McGraw-Hill, 1999.

KNOB, Ronan; SCHREINER, Geomar; FROZZA, Angelo; MELLO, Ronaldo dos Santos. Uma Análise de Soluções NewSQL. *In*: ESCOLA REGIONAL DE BANCO DE DADOS (ERBD), 15. , 2019, Chapecó. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019 . p. 21-30.

OLIVEIRA, Moacir; MELO, Nicole; SANTOS, Leandro; OLIVEIRA, Wellington. BANCO DE DADOS NO-SQL X BANCO DE DADOS SQL: ESTUDO DE DESEMPENHO EM GRANDES MASSAS. South American Development Society

Journal, volume 04, número 11, p. 298 - 320, agosto, 2018.

PAVLO, Andrew; ASLETT, Matthew. What´s Really New with NewSQL?. SIGMOD Record, volume 42, número 2, p.45 - 55, junho, 2016.

STONEBRAKER, M. Newsql: An alternative to nosql and old sql for new oltp apps.Communications of the ACM. Retrieved, p. 07–06, 2012.

STRAUCH, Christof. NoSQL Databases. Software and Systems Modeling, fevereiro, 2016.