



ANÁLISE DE DESEMPENHO ENTRE BANCO DE DADOS RELACIONAL (MYSQL) E NÃO-RELACIONAL (AMAZON DYNAMODB)

PILOTO LOURENÇO, Bárbara Raquel¹

JANUÁRIO, Nicolas dos Santos²

MASSARO JÚNIOR, Flávio Rubens³

Centro Universitário Hermínio Ometto – UNIARARAS, Araras – SP, Brasil

Resumo

Um banco de dados, é uma coleção de informações estruturadas, ou dados, normalmente armazenados eletronicamente em um sistema de computador. Alguns bancos de dados que não são classificados como relacional são definidos como NoSQL. Dado que esta tecnologia é nova e tem capacidade de trabalhar com grandes volumes de dados, é importante avaliar seu potencial alcance de aplicação em comparação com o modelo convencional de correlação, que vem sendo utilizado há décadas. Realizando operações de manipulação de dados em inúmeros volumes de transações simultâneas e comparando o desempenho de cada sistema, busca-se comparar neste trabalho os dois conceitos de banco de dados. Os sistemas utilizados foram MySQL (relacional) e DynamoDB (NoSQL). A análise foi realizada por meio de um script que rastreou cronometricamente o tempo de execução de cada um, a avaliação de desempenho foi realizada a partir da comparação dos tempos de respostas obtidos em cada execução e demonstrados por meio de tabela. Ao realizar os testes foi possível confirmar que o modelo NoSQL de forma geral mostrou um melhor desempenho no quesito tempo de execução.

Palavras chave: Análise Comparativa, Manipulação de dados, Banco de Dados, MySQL e DynamoDB.

¹ FHO|UNIARARAS. Bárbara Raquel Piloto Lourenço, barbarapiloto@alunos.fho.edu.br



² FHO|UNIARARAS. Nicolas dos Santos Januario, nicolas.janu@alunos.fho.edu.br

³ FHO|UNIARARAS. Flávio Rubens Massaro Júnior, frmassaro@fho.edu.br

1 Introdução

O volume de dados gerados e a necessidade de uso de SGBDs (Sistemas de Gerenciamento de Bancos de Dados) crescem a cada dia. Junto com essa demanda, cresce o volume de dados gerados por esses sistemas, assim como aumenta a complexidade de suas aplicações. Existem bancos (relações) em vez de bancos relacionados no espaço SGBDs que representam dados em coleções de documentos.

A opção mais adequada para uma aplicação em um ERP, CRM ou sistema de gestão financeira é usar um banco de dados relacional, devido a exigência de um alto grau de consistência de dados. O modelo foi criado em 1970 por Edgar Frank Codd e é o substituto do modelo hierárquico e rede.

A principal distinção entre modelos de bancos de dados relacionais e bancos de dados não relacionais é que em bancos de dados relacionais, os dados são mostrados em tabelas, colunas, linhas ou representações gráficas para fins de organização e recuperação de dados. Enquanto os não relacionados não contam com essas estruturas, sua modelagem de dados é mais flexível, sendo utilizado a organização por índice ou chaves.

Para realização deste projeto, serão utilizados: o banco de dados relacional MySQL e o não-relacional DynamoDB. Para permitir uma análise comparativa entre estes modelos foram criados três cenários de dados fictícios para que sejam traçadas métricas de seus desempenhos. Essa análise também analisou as operações com os modelos de banco de dados utilizando diferentes volumes de dados e de transações simultâneas. E por fim, ilustrando os resultados em gráficos e tabelas e obtendo questões relevantes para auxílio na implementação de cada banco de dados.

1.1 Contextualização

Este capítulo aborda uma explicação da área de pesquisa do trabalho. Além disso contextualizar o desenvolvimento envolvendo banco de dados.

1.2 Tema de Pesquisa



O tema deste trabalho aborda uma análise de desempenho entre banco de dados relacional e não relacional, para que fique detalhado qual o modelo é mais indicado para o fluxo de quantidade de dados simultâneos, com isso, permitindo a escolha do modelo mais adequado para utilização.

1.3 Motivações e Justificativas

A análise comparativa de desempenho permite a identificação de qual modelo é mais indicado em função do volume de transações simultâneas, permitindo também auxiliar na tomada de decisão a respeito de qual modelo deve ser utilizado para determinada aplicação.

A comparação entre estes bancos de dados visa esclarecer qual o melhor modelo de banco de dados para diferentes aplicações.

1.4 Objetivos

O objetivo geral deste trabalho é comparar os desempenhos entre os modelos de um Sistema de Gerenciamento de Banco de Dados relacional e um não relacional. Os objetivos específicos são:

- Ilustrar os resultados obtidos via tabelas;
- Criar cenários para realização dos benchmarks;
- Analisar o desempenho de dados juntamente com o desempenho;
- Criar um modelo de benchmark para comparação entre os bancos de dados MySQL e DynamoDB.

2 Revisão Bibliográfica

Nessa seção serão abordados os conceitos relacionados e trabalhos relacionados.

2.1 Conceitos Relacionados

Nesta seção serão apresentados os seguintes conceitos: Sistema de Gerenciamento de Banco de Dados, Banco de Dados Relacional, Banco de Dados Não-Relacional e Modelo Chave-Valor.

2.1.1 Sistemas de Gerenciamento de Bancos de Dados



Segundo C. J. Date(2004), um sistema de banco de dados é basicamente um sistema computadorizado que realiza manutenção de registros. Podemos imaginá-lo como um armário de arquivos, ele é um espaço para um conjunto de arquivos de dados digitalizados. Os usuários que fazem o uso desses sistemas podem realizar diversas operações, como inserção de novos dados, busca de dados, exclusão de dados, alteração de dados e remoção de dados.

Os Sistemas de Banco de Dados (SGBD) surgiram com o intuito de contornar os problemas apresentados no formato de armazenamento em arquivos de sistema operacional e são formados por uma série de componentes no nível de hardware e de software, dentre os quais se destaca o SGBD, que permite o gerenciamento de informações armazenadas em um Banco de Dados (DATE, 2003). Os SGBDs devem possuir uma série de características para evitar alguns problemas, tais como redundância e inconsistência dos dados, acesso indevido e problemas de atomicidade.

2.1.2 Banco de Dados Relacionais

O modelo de dados relacional foi introduzido por Codd (1970). Entre os modelos existentes de implementação, o modelo relacional é o mais simples, com uma estrutura de dados uniforme e formal.

Sua estrutura é formada por tabelas para representar dados e suas relações. Possui uma coleção de operadores, que constituem a base da linguagem SQL, é uma coleção de restrições de integridade, que definem um conjunto consistente de estados de base de dados e de alterações de estados (SILBERSCHATZ; KORTH; SUDARSHAN, 2006).

Segundo Takai (2005), o banco de dados do modelo relacional representa os dados do banco em uma coletânea de relações, onde cada relação pode ser interpretada em forma de tabela e arquivos de registro.

2.1.2.1 MySQL

O MySQL é um servidor e SGBD relacional desenvolvido para aplicações de pequeno e médio porte, hoje atende aplicações de grande porte, tornando assim com mais



vantagens do que seus concorrentes. MySQL vem sendo reconhecido como um banco de dados *open Source* com capacidade de concorrer programa de código fechado tais como SQL Server e Oracle (MILANI, 2006). O mesmo utiliza a linguagem SQL e, mesmo diante de um constante avanço em tecnologias que utilizam outros paradigmas, é um dos Bancos de Dados mais populares devido à sua facilidade de uso e confiabilidade (MYSQL, 2012).

2.1.3 Banco de Dados Não-Relacionais

Segundo Garcia (2019), os SGBDs NoSQL, apresentam algumas características fundamentais que os diferenciam dos relacionais, fazendo deles adequados para armazenamento de grandes volumes de dados não estruturados. Como por exemplo Escalabilidade Horizontal, Ausência de esquema flexível, Suporte nativo a replicação, API simples para acesso aos dados, Consistência eventual.

Existem vários modelos de bancos de dados não-relacionais, sendo eles o chave-valor, que consiste em uma tabela hash simples, utilizada quando todo o acesso ao banco de dados é feito por meio da chave primária. O modelo documento, em que o conceito principal é armazenar e recuperar documentos, os quais podem ser XML, JSON, BSON, entre outros. O modelo orientado a grafos, que tem como ideia armazenar, mapear e consultar relacionamentos em teoria de grafos. São basicamente uma coleção de nós, onde um nó é igual uma entidade e suas arestas são a representação do relacionamento entre nós. E o modelo colunar, que possui o seu armazenamento orientado a colunas, que tem um ganho significativo em sua performance, pois diminuem a quantidade de dados que o disco precisa carregar (PRAMOD, 2019)

2.1.3.1 Chave Valor

O banco de dados de chave-valor é um tipo de banco de dados não relacional que usa um método de chave-valor simples (DA COSTA, 2016). O banco de dados de chave-valor funciona como um conjunto de pares em que a chave-valor funciona como um identificador exclusivo, o banco de chave-valor são particionáveis e permitem



escalabilidade horizontal que facilita de lidar com um número maior de solicitações de transações.

2.1.3.2 Documentos

O Banco de Dados orientado a documentos é baseado no armazenamento de pares de chave-valor tendo um esquema bem flexível, e esta qualidade o torna o banco de dados orientado em uma ótima escolha para dados com uma estrutura que não está de acordo com os modelos formais. O modelo orientado tem como um conjunto de documentos e cada documento tem os seus campos de pares de chave-valor e este modelo não depende de um esquema rígido então ele não necessita de uma estrutura padrão como ocorre em banco de dados relacionais.(OLIVEIRA S. S, 2014).

2.1.3.3 Amazon DynamoDB

O DynamoDB é um banco de dados não-relacional do tipo chave-valor e documentos totalmente gerenciado, escalonável para Internet, de fácil administração e baixo custo. É uma parte do painel de serviços oferecidos pela Amazon. Foi construído considerando uma escalabilidade rápida e contínua de alto desempenho. É executado em SSDs para fornecer respostas mais rápidas e não tem limites de capacidade e solicitação de armazenamento (DESHPANDE, 2014).

2.1.4 Apache Jmeter

Apache JMeter™ é um software de código aberto, um aplicativo Java 100% puro projetado para carregar o comportamento funcional de teste e medir o desempenho. Ele foi originalmente projetado para testar aplicativos da Web, mas desde então se expandiu para outras funções de teste. O Apache JMeter pode ser usado para testar o desempenho em recursos estáticos e dinâmicos, aplicativos dinâmicos da Web. Ele pode ser usado para simular uma carga pesada em um servidor, grupo de servidores, rede ou objeto para testar sua força ou analisar o desempenho geral sob diferentes tipos de carga (JMETER APACHE, 2022).

2.2 Trabalhos Relacionados



Takai, Italiano e Ferreira (2005), sobre Introdução a Banco de Dados, explicam o que são sistemas de gerenciamento de banco de dados, detalhando sobre os modelos (Hierárquico, Rede, Relacional, Orientado a Objetos, Objetos-Relacionais), arquiteturas, capacidades, modelo de dados, esquema e instâncias, conceitos do modelo entidade-relacionamento (entidades e atributos, tipos de entidades, conjunto de valores e atributos-chaves, relacionamentos, papéis e restrições estruturais, tipo de entidade-frac), linguagens formais de consulta, linguagem SQL, dependências funcionais e normalização de base de dados relacionais, Data Warehouse, porém não apresenta explicativa entre esses bancos de dados e nenhum benchmark entre os mesmos.

Já Silva e Ferreira, (2017), em seu trabalho sobre análise comparativa de desempenho de consultas entre um banco de dados relacional e um não-relacional, foi utilizado uma base de dados da Microsoft *AdventureWorks* onde a base já vem toda estruturada. Seu banco de dados relacional escolhido para o benchmark foi o SQL Server e o não-relacional o MongoDB. Para a comparação foi criado um software em *.net* na plataforma Microsoft Visual Studio 2015 onde os scripts são colocados lado a lado e o usuário pode escolher o tipo de busca que será realizada em ambas as bases de dados.

Aquino (2019), em sua análise comparativa de sistemas de gerenciamento de bancos de dados NoSQL multimodelo tem como principal objetivo compará-los em termos de suporte à administração, manipulação e armazenamento de dados. Seu trabalho foi feito em forma de pesquisa, sem uma análise de desempenho entre eles e somente fazendo o uso de bancos de dados não-relacionais.

Godoy e Pinheiro (2021), em seu trabalho de Análise de desempenho entre banco de dados relacional (MySQL) e não relacional (MongoDB), sua base de dados vem toda estruturada por Farias (2014) e Cuer (2014), suas comparações são em tabelas de campos simples e compostos.

3 Metodologia

No projeto, foram realizadas simulações a fim de obter o tempo de resposta em cada *thread*, considerando sua realização em ambos bancos de dados. Apresentando os resultados obtidos via tabelas, de forma clara e objetiva que foram utilizados para a



identificação do modelo que apresentou o melhor desempenho em relação à complexidade da busca de informações.

Este trabalho consiste em realizar um grupo de testes de desempenho em banco de dados relacional e banco de dados não-relacional, a fim de analisar o comportamento destes SGBDs para sistemas que armazenam diversas quantidades de informações em seus bancos de dados em diferentes volumes de transações simultâneas.

O projeto foi executado por meio de 6 fases: Preparação de Estudo, Preparação de ambiente, modelagem, desenvolvimentos, testes e análise dos resultados.

3.1 Preparação de Ambiente

Para realizar a análise clara e sem interferência optou-se por configurar dois ambientes de testes virtuais utilizando o Oracle VM Virtualbox, em que foram instaladas apenas ferramentas obrigatórias e necessárias para o teste de cada SGBD. Os sistemas contém o mesmo poder computacional para que haja uma análise justa.

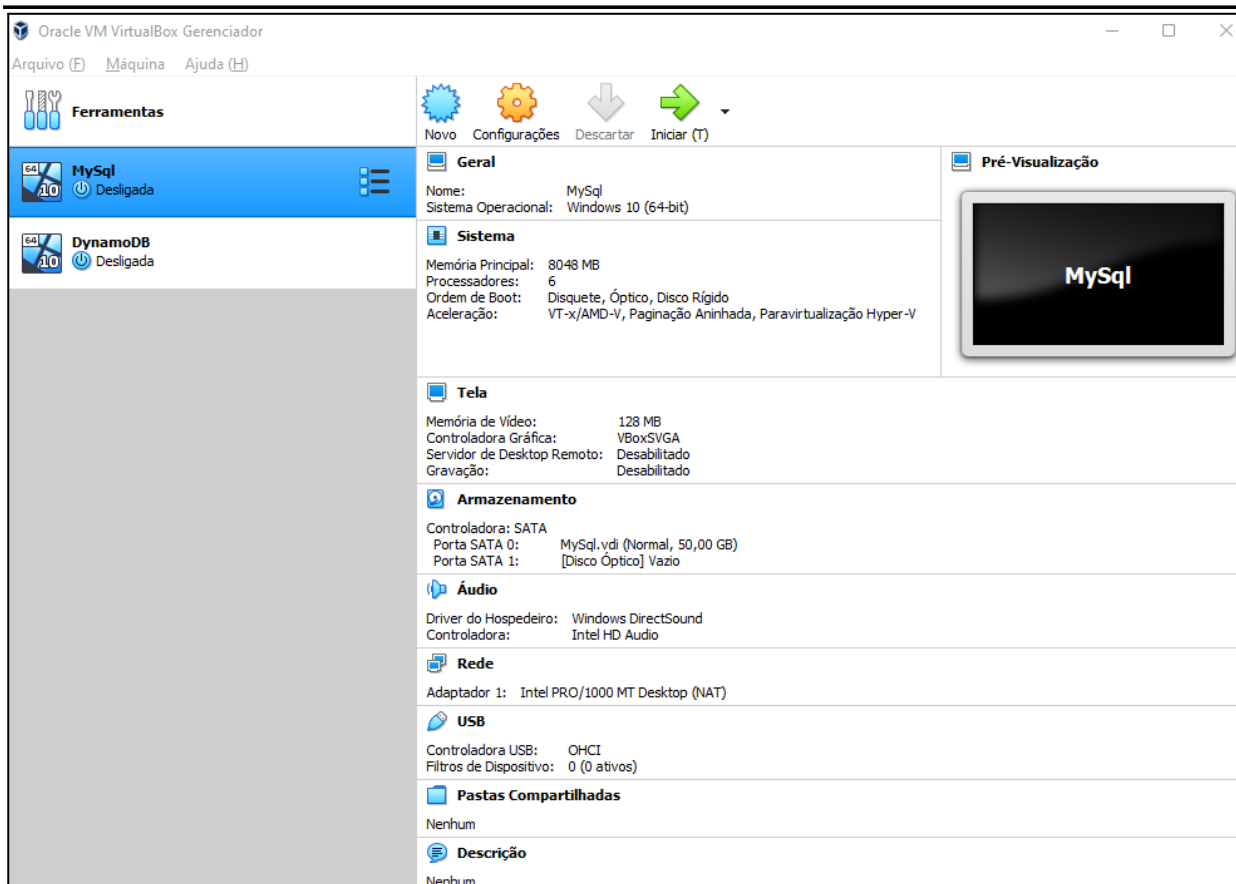
3.1.1 Máquina Host

A máquina hospedeira a ser utilizada é um desktop que conta com um processador AMD Ryzen 5 1600 6-core, placa de vídeo Nvidia 1650 com 4 GB de memória, 16 GB de memória RAM ddr4, disco rígido de 1TB e SSD com 111GB. O sistema operacional é o Windows 11 Pro 64 bits.

3.1.2 Ambiente Virtual de Testes

Cada ambiente virtual contou com 8 GB de RAM, 6 núcleos do processador e um SSD de 50GB dinamicamente alocado. Devido à familiaridade de utilização e por ser fornecido gratuitamente pela FHO, o sistema operacional a ser utilizado será o Windows Education 64 bits.(Figura 1).

Figura 1 - Especificações da VM

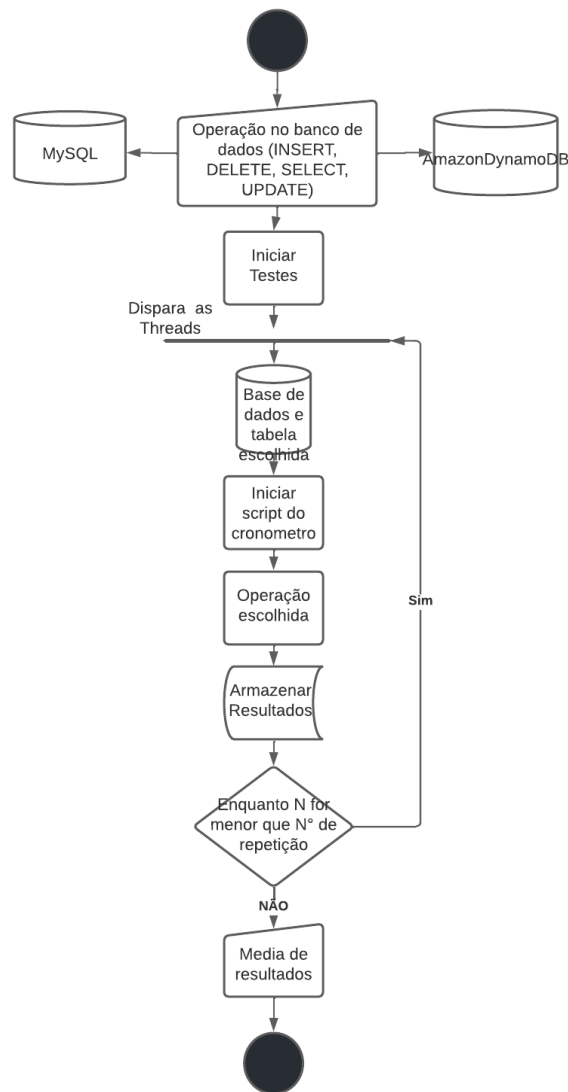


Fonte: Elaborado pelos autores

3.2 Modelagem

Esta fase inclui o processo de modelagem do projeto. Essa etapa utilizou a ferramenta LucidChart para criar um fluxograma (Figura 2) do sistema que mostra a sequência de etapas em que o algoritmo realizou o teste. Pode-se ver a configuração dos número de vezes que o processo selecionado foi executado e o banco de dados que foi utilizado, e um modelo da estrutura do banco de dados e das operações que serão realizadas no banco de dados. Após essa configuração, o teste foi iniciado, a *thread* foi liberada, através do banco de dados selecionado, e por fim, o script de teste foi executado para determinar a média dos resultados. Ao iniciar a operação selecionada, o script salva o tempo após o término da operação e retorna o tempo médio gasto na operação. A base de dados que foram utilizadas são bases prontas na linguagem Json, pois ambos os bancos são compatíveis e para que haja uma melhor veracidade de dados

Figura 2 - Fluxo de testes de dados nos bancos

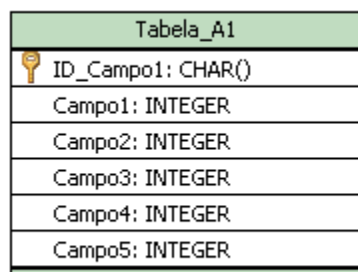


Fonte: Elaborado pelos autores

O brModelo foi utilizado para a criação do modelo lógico do banco de dados, que conta com 2 modelos com diferentes quantidades de relacionamentos que serão utilizados para simular diferentes aplicações.

A Figura 3 mostra o Modelo Lógico A, desenvolvido para simular uma base de dados que não possui relacionamentos.

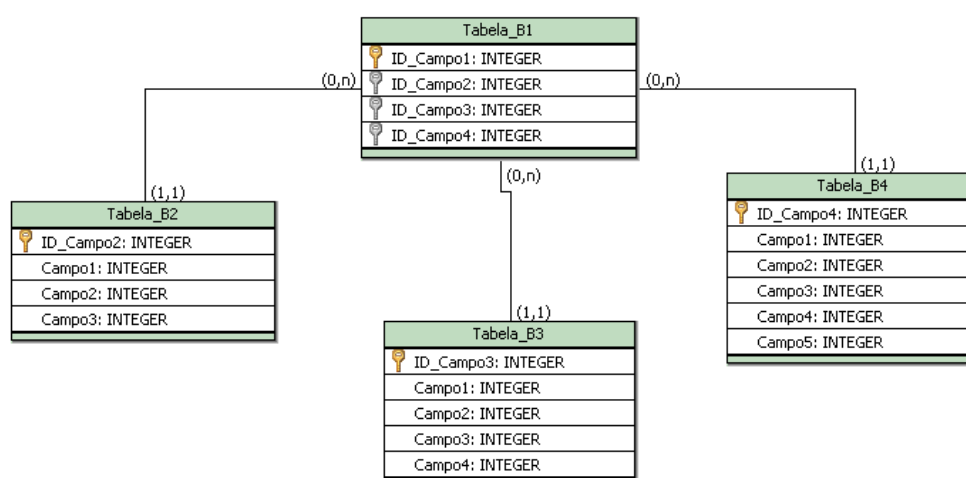
Figura 3 - Modelo lógico A



Fonte: Elaborado pelos autores

A Figura 4 mostra o modelo lógico B, desenvolvido para simular uma base de dados de com três relacionamentos.

Figura 4 - Modelo lógico B



Fonte: Elaborado pelos autores

Foram utilizados os mesmos 2 modelos para criação da modelagem NoSQL, que foi feita seguindo o formato JSON. Os campos compostos em um modelo JSON são equivalentes aos relacionamentos do modelo relacional.

A Figura 5 mostra o Modelo JSON A, desenvolvido para simular uma base de dados que não possui campos compostos.

Figura 5 - Modelo JSON A

```
Tabela_A1{
  _id:<ObjectId>,
  ID_Campo1: int,
  ID_Campo2: int,
  ID_Campo3: int,
  ID_Campo4: int
}
```

Fonte: Elaborado pelos autores

A Figura 6 mostra o Modelo JSON B, foi desenvolvido para simular uma base de dados com três campos compostos.

Figura 6 - Modelo JSON B

```
Tabela_B1
{
  _id:<ObjectId>,
  Tabela_B2:{
    Campo1: int,
    Campo2: int,
    Campo3: int
  },
  Tabela_B3:{
    Campo1: int,
    Campo2: int,
    Campo3: int,
    Campo4: int
  },
  Tabela_B4:{
    Campo1: int,
    Campo2: int,
    Campo3: int,
    Campo4: int,
    Campo5: int
  }
}
```

Fonte: Elaborado pelos autores

3.3 Desenvolvimento

Na fase de desenvolvimento, primeiramente foram definidas as ferramentas para implementação da modelagem realizada anteriormente.

3.3.1 Processo de Seleção dos SGBDs Objetos Deste Estudo

A escolha dos SGBDs teve como base o site DB-Engines (Figura 7) que forma um ranking de popularidade, também foi utilizado como critério que os softwares deveriam ser gratuitos e com facilidade de acesso.



Com isso, os SGBDs com as melhores colocações foram: o MySQL para o modelo relacional e o Amazon DynamoDB para o modelo não relacional.

Figura 7 - DB-Engines Ranking

Classificação de motores de banco de dados

O DB-Engines Ranking classifica os sistemas de gerenciamento de banco de dados de acordo com sua popularidade. O ranking é atualizado mensalmente.

Leia mais sobre o [método](#) de cálculo das pontuações.



395 sistemas no ranking, setembro de 2022

Classificação			SGBD	Modelo de banco de dados	Pontuação		
Setembro de 2022	agosto de 2022	setembro de 2021			Setembro de 2022	agosto de 2022	setembro de 2021
1.	1.	1.	Oracle+	Relacional, Multimodelo	1238,25	-22,54	-33,29
2.	2.	2.	MySQL+	Relacional, Multimodelo	1212,47	+9,61	-0,06
3.	3.	3.	Microsoft SQL Server+	Relacional, Multimodelo	926,30	-18,66	-44,55
4.	4.	4.	PostgreSQL+	Relacional, Multimodelo	620,46	+2,46	+42,95
5.	5.	5.	MongoDB+	Documento, Multimodelo	489,64	+11,97	-6,87
6.	6.	6.	Redis+	Valor-chave, vários modelos	181,47	+5,08	+9,53
7.	8.	8.	Elasticsearch	Motor de busca, multimodelo	151,44	-3,64	-8,80
8.	7.	7.	IBM DB2	Relacional, Multimodelo	151,39	-5,83	-15,16
9.	9.	11.	Microsoft Access	Relacional	140,03	-6,47	+23,09
10.	10.	9.	SQLite+	Relacional	138,82	-0,05	+10,17
11.	11.	10.	Cassandra+	Coluna larga	119,11	+0,97	+0,12
12.	12.	12.	MariaDB+	Relacional, Multimodelo	110,16	-3,74	+9,46
13.	13.	21.	Floco de neve+	Relacional	103,50	+0,38	+51,43
14.	14.	13.	Splunk	Motor de busca	94,05	-3,39	+2,45
15.	15.	16.	Amazon DynamoDB+	Vários modelos	87,42	+0,16	+10,49

Fonte: DB-Engines (2022)

3.3.2 Definição da Linguagem de Programação a Ser Utilizada nos Testes

Foram utilizados o framework o JMeter é uma ferramenta desktop gratuita de código aberto feita em Java que foi desenvolvida para realizar testes de performance.

3.3.3 Testes

Nesta fase, os parâmetros para realização do teste em cada banco de dados. Determina-se para cada complexidade respectivamente: o número de acessos simultâneos (*threads*) foram de: 1, 2, 5, 10, 50; já os número de registros manipulados foram: 100, 500, 1000 e 2000 e foi necessário repetir 5 vezes cada caso, para que houvesse maior exatidão no tempo de resposta em cada banco. Os números de registros



manipulados foram baseados nos trabalhos de Godoy (2021) e Pinheiro (2021) com adaptação necessária para compatibilidade das *threads*.

3.4 Análise dos resultados

Os resultados obtidos anteriormente foram analisados por meio de desvio padrão e da média do tempo de execução em milissegundos de cada caso pelo número de vezes em que foi testado e depois ilustrado em tabelas para comparação entre os dois modelos.

4 Resultados

Nesta seção encontram-se os resultados dos testes realizados nos bancos de dados MySQL e Amazon DynamoDB em cada um dos dois modelos desenvolvidos anteriormente. As tabelas abaixo mostram os tempo de execução de cada operação de cada banco de dados para cada modelo desenvolvido, sendo eles A e B e os resultados são exibidos em milissegundos, na qual as colunas da tabela representam os números de acessos simultâneos e as linhas mostram a quantidade de registros manipulados na comparação. Nas tabelas abaixo a cor verde representa o menor tempo de execução.

4.1 INSERT

Para os testes de *INSERT* os dados inseridos seguiram um padrão de números crescentes a partir do zero e todos os campos das tabelas com os mesmos valores. Resultados dos modelos A e B estão indicados nas tabelas 1 e 2 respectivamente.

Tabela 1 - Comparação *Insert* no modelo A

MySQL							DynamoDB						
Modelo A	1	2	5	10	50	100	Modelo A	1	2	5	10	50	100
100	0,31	0,28	0,16	0,13	0,49	0,54	100	0,09	0,14	0,18	0,43	0,83	1,05
1000	2,33	2,29	1,22	0,63	1,35	1,37	1000	0,12	0,13	0,15	0,23	0,46	1,13
10000	27,55	22,11	8,68	4,91	2,19	2,24	10000	0,16	0,2	0,24	0,43	0,89	0,87
100000	249,02	240,65	100,02	63,1	18,04	11,95	100000	0,93	0,780	0,65	0,47	0,95	1,54

Fonte: Elaborado pelos autores



No modelo A, o *Amazon DynamoDB*, mostrou ser mais rápido na maioria dos casos com um diferencial conforme o aumento de registro, o melhor tempo ao trabalhar com 100 registros a partir dos 5 acessos simultâneos

Tabela 2 - Comparação *Insert* no modelo B

MySQL							DynamoDB						
Modelo B	1	2	5	10	50	100	Modelo B	1	2	5	10	50	100
100	1,15	1,13	1,18	0,59	1,09	3,08	100	0,09	0,14	0,24	0,39	0,77	1,3
1000	9,97	9,63	4,32	2,48	1,27	1,68	1000	0,1	0,13	0,25	0,43	0,84	0,98
10000	111,33	91,21	36,39	20,56	8,55	6,01	10000	0,17	0,2	0,19	0,36	0,8	1,39
100000	1231,29	1019,09	425,63	225,44	84,72	71,32	100000	1,6	1,090	0,071	0,51	1,5	2,35

Fonte: Elaborado pelos autores

No modelo B, o Amazon DynamoDB teve um tempo de execução menor em todas as intenções.

4.2 SELECT

Para os testes de SELECT foram selecionados os campos das tabelas, incluindo os devidos relacionamentos necessários para cada modelo. Resultados dos modelos A e B estão indicados nas tabelas 3 e 4 respectivamente.

Tabela 3 - Comparação *Insert* no modelo A

MySQL							DynamoDB						
Modelo A	1	2	5	10	50	100	Modelo A	1	2	5	10	50	100
100	0,055	0,063	0,109	0,17	1,522	4,373	100	0,031	0,033	0,041	0,101	1,234	1,373
1000	0,053	0,057	0,094	0,161	1,682	5,421	1000	0,043	0,042	0,073	0,103	1,482	3,421
10000	0,054	0,061	0,109	0,179	3,633	7,343	10000	0,045	0,053	0,081	0,139	1,633	4,343
100000	0,835	1,035	1,133	1,323	13,142	17,531	100000	0,735	0,901	1,121	1,225	8,142	18,531

Fonte: Elaborado pelos autores

No modelo A, o MySQL apresentou um melhor desempenho em uma caso específico ao se tratar de 100.000 registros e 100 acessos simultâneos.



Tabela 4 - Comparação Insert no modelo B

MySQL							DynamoDB						
Modelo B	1	2	5	10	50	100	Modelo B	1	2	5	10	50	100
100	0,054	0,061	0,104	0,171	1,622	5,373	100	0,045	0,035	0,095	0,101	1,142	3,373
1000	0,058	0,059	0,107	0,173	1,704	5,721	1000	0,049	0,043	0,084	0,104	1,472	4,427
10000	0,055	0,066	0,109	0,182	3,601	7,343	10000	0,048	0,057	0,981	0,143	2,653	6,532
100000	0,962	1,125	1,198	1,341	33,147	57,531	100000	0,809	0,991	1,253	1,353	9,142	37,531

Fonte: Elaborado pelos autores

No modelo B, novamente o MySQL apresentou um melhor desempenho em um caso específico ao se tratar com 100.000 e 5 acessos simultâneos.

4.3 UPDATE

Para os testes de *UPDATE* foram atualizados somente o primeiro campo de cada tabela, levando em consideração os relacionamentos também foi somente atualizado o primeiro campo das tabelas filhas. O valor utilizado na atualização é uma constante em todos os modelos. Resultados dos modelos A e B estão indicados nas tabelas 5 e 6 respectivamente.

Tabela 5 - Comparação Insert no modelo A

MySQL							DynamoDB						
Modelo A	1	2	5	10	50	100	Modelo A	1	2	5	10	50	100
100	0,6	0,65	0,56	0,92	3,5	4,93	100	0,31	0,41	0,39	0,44	1,39	2,42
1000	2,71	2,68	1,59	1,46	3,33	5,01	1000	0,74	0,47	0,35	0,47	1,48	2,5
10000	24,3	23,94	10,69	6,03	5,51	4,93	10000	30,31	20,22	10,1	7,79	7,78	9,74
100000	271,45	251,41	101,2	63,57	24,82	23,07	100000	2813,41	1312,33	689,23	594,78	493,81	479,73

Fonte: Elaborado pelos autores

No modelo A, o Amazon DynamoDB apresentou um desempenho melhor ao operar com 100 e 1000 registros, porém o MySQL mostrou melhor desempenho com uma maior



quantidade de registros (a partir dos 10000), com uma única execução, sendo 10000 registros e 2 threads.

Tabela 6 - Comparação Insert no modelo B

MySQL							DynamoDB						
Modelo B	1	2	5	10	50	100	Modelo B	1	2	5	10	50	100
100	0,34	1,43	0,49	0,85	2,97	3,88	100	0,2	0,23	0,29	0,41	1,43	2,35
1000	2,4	2,53	2,63	3,21	6,43	6,98	1000	2,93	1,65	1,13	1,37	1,78	3,13
10000	173,76	171,4	170,94	174,44	178,18	187,19	10000	160,41	87,41	46,1	43,66	42,59	51,53
100000	16140,44	16090,1	15400,92	16002,51	16245,9	16001,34	100000	15231,42	9.234,130	4214,51	4123,45	4121,62	4014,68

Fonte: Elaborado pelos autores

No modelo B, o Amazon DynamoDB demonstrou um melhor desempenho com um menor tempo de execução, exceto o único caso com 1000 registros e 1 acesso.

4.4 DELETE

Para os testes de DELETE, foram executados linha por linha, identificando o registro pelo id de cada linha e começando a exclusão pela tabela que recebia os relacionamentos e por sequência as demais. Resultados dos modelos A e B estão indicados nas tabelas 7 e 8 respectivamente.

Tabela 7 - Comparação Insert no modelo A

MySQL							DynamoDB						
Modelo A	1	2	5	10	50	100	Modelo A	1	2	5	10	50	100
100	0,54	0,56	0,57	1,17	3,41	4,97	100	0,23	0,26	0,49	0,55	1,91	2,5
1000	2,65	2,54	1,33	1,31	3,54	4,21	1000	0,46	0,45	0,25	0,63	1,01	1,86
10000	24,64	22,25	10,08	6,64	5,11	8,23	10000	2,87	2,93	3,13	3,09	4,52	6,98
100000	245,71	229,66	95,74	52,61	23,07	25,41	100000	29,02	17,08	15,27	42,56	213,2	225,91

Fonte: Elaborado pelos autores



No modelo A, nota-se uma vantagem do Amazon DynamoDB na grande maioria dos números de relacionamento e acessos, exceto nos casos de 100.000 registros com 50 e 100 acessos.

Tabela 8 - Comparação Insert no modelo B

MySQL							DynamoDB						
Modelo B	1	2	5	10	50	100	B	1	2	5	10	50	100
100	1,42	3,21	0,85	1,54	3,86	8,55	100	0,15	0,18	0,25	0,38	1,74	2,37
1000	10,88	9,54	4,65	3,48	3,94	5,11	1000	0,32	0,26	0,28	0,44	1,65	2,97
10000	104,66	90,03	38,46	21,15	8,33	8,94	10000	2,87	2,2	2,09	3,63	4,28	5,98
100000	1276,62	1196,15	397,52	211,94	60,55	102,21	100000	29,14	25,340	19,29	45,77	183,52	225,97

Fonte: Elaborado pelos autores

No modelo B observa-se que os casos que obtiveram o menor tempo de resposta, foram idênticos ao da tabela anterior.



5 Considerações Finais

Os bancos de dados NoSQL estão ganhando popularidade devido à sua capacidade de lidar com grandes quantidades de dados. Por se tratar de uma tecnologia nova, é fundamental examinar seu escopo de aplicação em comparação com o modelo relacional que vem sendo utilizado há décadas.

O presente trabalho mostrou uma análise no tempo de execução entre dois SGBDs com tecnologias de banco de dados diferentes, sendo elas: o modelo Relacional (MySQL) e NoSQL (Amazon DynamoDB), aplicando diferentes volumes de dados e transações simultâneas.

Usando em consideração nesta modelagem os resultados deixaram uma clara vantagem para o Amazon DynamoDB sobre o MySQL nas operações de INSERT, DELETE, já na operação de SELECT teve uma similaridade, e na operação de UPDATE o MySQL obteve um melhor resultado, sendo ela no modelo A, tendo um equilíbrio nos outros modelos.

Apesar de o escopo desse projeto ter em foco analisar o tempo de execução de cada SGBD, é válido ressaltar que os bancos de dados são usados não apenas por sua velocidade mas também por sua integridade.

Assim fica como sugestão para trabalhos futuros uma comparação de desempenho com o foco na integridade dos dados.



Referências Bibliográficas

Amazon DynamoDB. **Definição de banco de dados de chave-valor**. Disponível em: < <https://aws.amazon.com/dynamodb/> > Acesso em: 07 de Outubro de 2021.

Apache JMeter. **Apache JMeter™**. Disponível em: <<https://jmeter.apache.org/>>. Acesso em: 21 de Outubro de 2022

Aquino A.C. **Uma Análise Comparativa de Sistemas de Gerenciamento de Banco de Dados NoSQL Multimodelo**. UFSC 2019. 81p.

BARCELAR R.R. **Introdução ao estudo de banco de dados**, 2012. 71p.

BARROS J.R.A. et al. **Análise de Desempenho de Banco de Dados Relacionais e Não Relacionais em Dados Genômicos**. 2017. 17p.

DA COSTA, Augusto Verzbickas; VILAIN, Patricia; MELLO, Ronaldo dos Santos. **Uma Camada para o Mapeamento de Instruções SQL DML para o Banco de Dados NoSQL Chave-Valor Voldemort**. In: SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO (SBSI), 12., 2016, Florianópolis. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2016. p. 224-231.

DATE, C. J. **Uma Introdução a Sistemas de Banco de Dados**. 8. ed. Traduzido por Daniel Vieira. Rio de Janeiro: Elsevier, 2003.

DANTE, C. J. **Introdução a Sistemas de BANCOS de DADOS**. São Paulo: Elsevier, 2004. 251p.

DESHPANDE T. **Mastering DynamoDB**. 2014. 236p.

FERREIRA E.R. **Análise de Desempenho de Bancos de Dados**. Barbacena, 2019. 20p.

GARCIA V.S. E SOTTO E.C. **Comparativo entre os modelos de Banco de Dados Relacional e Não-Relacional**. Catanduva, 2019. 13p.

JUNIOR C. C. **SISTEMAS INTEGRADOS DE GESTÃO ERP UMA ABORDAGEM GERENCIAL**. 2012. 208p.

MACHADO F.N.R. **Banco de Dados: Projeto e Implementação**. São Paulo. 2020. 376p.
Milani André, MySQL Guia do Programador, Novatec Editora, 2006. 392p.

MySQL. Site oficial. Disponível em: <<https://www.mysql.com/>> . Acesso em: 06 Set. 2022



OLIVEIRA S.S. BANCOS DE DADOS NÃO-RELACIONAIS: UM NOVO PARADIGMA PARA ARMAZENAMENTO DE DADOS EM SISTEMAS DE ENSINO COLABORATIVO. 2014. 194p.

PRAMOD J.S. NOSQL ESSENCIAL: UM GUIA CONCISO PARA O MUNDO EMERGENTE DA PERSISTÊNCIA POLIGLOTA. 2019. 218p.

SILBERSCHATZ, Abraham; KORTH, Henry F; SUDARSHAN, S. **Sistema de banco de dados.** 5. ed. Traduzido por Daniel Vieira. Rio de Janeiro: Elsevier, 2006.

SILVA G.J.F. OLIVEIRA J.C. **Análise Comparativa de Desempenho de Consultas entre um Banco de Dados Relacional e um Banco de Dados Não Relacional.** 2017. 26p.

SILVA J.C. **Um estudo comparativo entre bancos de dados NoSQL: DynamoDB e Redis.** Itajubá. 2019. 8p.

SILVA G.J. FERREIRA J.C.O. **Análise Comparativa de Desempenho de Consultas Entre Um Banco de Dados Relacional e Um Banco de Dados Não-Relacional.** 2017. 26p.

TAKAI O.K. ITALIANO I.C. FERREIRA J.E. **Introdução a Banco de Dados.** DCC-IME-USP 2005. 124p.