



UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE CIÊNCIAS EXATAS
DEPARTAMENTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Matheus Mendes de Moraes Fraresso

Comparação de desempenho entre bancos relacionais e não relacionais

**Curitiba
2023**

Matheus Mendes de Moraes Fraresso

**Comparação de desempenho entre bancos relacionais e
não relacionais**

Trabalho de Conclusão de Curso apresentado do Curso de Graduação em Ciência da Computação da Universidade Federal do Paraná, como exigência parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Bruno Müller Júnior

**Curitiba
2023**

Agradecimentos

Dedico esse trabalho primeiramente aos que não tiveram a oportunidade de estar aqui hoje: Meu avô Paulo, minha avó Nilza e meu avô Divonsir, saibam que por mais que sua presença faça falta, sua lembrança e ensinamentos me acompanham todos os dias. Em seguida dedico este trabalho às pessoas que me ajudaram nessa jornada: Meus pais que sempre apoiaram, sempre sendo os primeiros a comemorar meus sucessos. Meus amigos que permitiram que eu não enfrentasse as dificuldades sozinho. Ao meu orientador, o professor Bruno Müller, que demonstrou paciência e compreensão muito além do que eu poderia pedir de qualquer pessoa na sua posição. E em especial à minha namorada Amanda, uma pessoa que me faltam palavras para descrever o altruísmo, moral, capacidade de confiar e sentir amor, saiba que esse trabalho só se completou por conta da sua fé inabalável em mim, mesmo quando eu mesmo não conseguia ter a mesma fé em mim

Resumo

Este trabalho consiste numa comparação do desempenho dos modelos relacional e não relacional dentro de um conjunto limitado de dados. Usando os sistemas gerenciadores de bancos de dados MongoDB e PostgreSQL, dados sintéticos foram usados para simular uma base de dados de cada modelo. Com as bases criadas, foi possível executar operações que tiveram seu tempo de execução medido e armazenado como métrica de avaliação para o desempenho do modelo. O modelo de dados usado foi descrito no trabalho, permitindo que os mesmos testes sejam reproduzidos em outros ambientes e a pesquisa continuada e expandida. Com esses dados foi possível gerar uma conclusão em relação a qual base de dados foi a mais efetiva para o cenário específico apresentado.

Palavras-chave: Banco de dados, Relacional, Não relacional, Comparação, Desempenho.

Sumário

1	INTRODUÇÃO	1
2	REVISÃO DE LITERATURA	4
2.1	Dados	4
2.2	Bancos de Dados	4
2.3	Bancos Relacionais	5
2.3.1	ACID	6
2.3.2	SQL	6
2.4	Bancos não relacionais	7
2.4.1	Sistemas distribuídos e CAP	8
3	DESCRIÇÃO CONCEITUAL	10
3.1	SGBDs	10
3.2	SGBDs	11
3.3	Algoritmo, software e hardware	11
3.4	Procuras	12
3.5	Métodos de avaliação	13
3.6	Expectativas	14
4	RESULTADOS E DISCUSSÃO	15
4.1	O Modelo	15
4.2	Dados sintéticos	16
4.3	Processamento de dados e operações	16
4.4	Testes efetuados	16
4.5	Resultados obtidos	17
5	CONCLUSÃO	27
	REFERÊNCIAS	29

Lista de ilustrações

Figura 1 – Diagrama de Modelo Relacional	15
Figura 2 – Dados coletados na ferramenta Planilhas do google	17
Figura 3 – Gráfico de tempo de execução de operações CREATE no SGBD MongoDB.	18
Figura 4 – Gráfico de tempo de execução de operações READ no SGBD MongoDB.	19
Figura 5 – Gráfico de tempo de execução de operações UPDATE no SGBD MongoDB.	19
Figura 6 – Gráfico de tempo de execução de operações DELETE no SGBD MongoDB.	20
Figura 7 – Gráfico de tempo de execução de operações recuperando Todos os dados no SGBD MongoDB.	20
Figura 8 – Gráfico de tempo de execução de operações 1:N no SGBD MongoDB.	21
Figura 9 – Gráfico de tempo de execução de operações N:N no SGBD MongoDB.	21
Figura 10 – Gráfico de tempo de execução de operações CREATE no SGBD PostgreSQL.	22
Figura 11 – Gráfico de tempo de execução de operações READ no SGBD PostgreSQL.	23
Figura 12 – Gráfico de tempo de execução de operações UPDATE no SGBD PostgreSQL.	23
Figura 13 – Gráfico de tempo de execução de operações DELETE no SGBD PostgreSQL.	24
Figura 14 – Gráfico de tempo de execução de operações recuperando Todos os dados no SGBD PostgreSQL.	24
Figura 15 – Gráfico de tempo de execução de operações 1:N no SGBD PostgreSQL.	25
Figura 16 – Gráfico de tempo de execução de operações N:N no SGBD PostgreSQL.	25

1 Introdução

O homem é o único animal que pode se comunicar por meio de símbolos abstratos (Hockett, Charles F., and Charles D. Hockett, 1960), isso significa que o ser humano é o único animal que sentiu a necessidade de transformar a informação anteriormente comunicada com sons, primeiramente em desenhos e depois em símbolos específicos que representam como conseguem emitir. Após anos de evolução esses símbolos deram origem às letras do nosso alfabeto, que foram a base fundamental para a criação da linguagem escrita. A linguagem escrita moderna tem características interessantes e muito diferentes da linguagem falada, por exemplo:

- **Persistência:** Após ser criada, uma informação escrita não depende mais do seu criador para ser comunicada, permitindo sua propagação para gerações futuras.
- **Precisão:** Por seguir padrões mais rígidos, as ideias podem ser transmitidas com menos erros, redundâncias e ambiguidades.
- **Conveniência:** A linguagem escrita pode ser acessada quando for mais conveniente, ou no momento que for mais necessária.
- **Repetibilidade:** Estando escrita, uma informação pode ser revista com precisão o número de vezes que for necessário, sem mudanças de contexto ou conteúdo.

Com a criação da linguagem escrita as pessoas passaram a transmitir suas ideias em textos, e o aumento da população também fez com que cada vez mais textos fossem criados, necessitando então de uma maneira de armazená-los e posteriormente recuperá-los para consulta. A solução foi a criação das bibliotecas, e acredita-se que umas das primeiras tenha sido a Biblioteca de Alexandria, fundada em 3 A.C ¹ indicando que o ser humano procura armazenar e disponibilizar informação há milênios. Até esse momento da história, a hierarquia da informação tinha um formato aproximado a esse: letras geram palavras, que geram linhas, que formam parágrafos, que compõem páginas, que formam livros (ou tábuas) que devem ser guardados em estantes, e várias dessas formam acervos e bibliotecas. E esse formato de guardar informação se manteve bastante estável até o século XX, quando houve avanços na matemática e elétrica que permitiram que o modo com que guardamos informação tivesse uma grande revolução, que foi a criação de computadores e da informação digital. Com esse novo formato de informação, pudemos aumentar muito nossa capacidade de gerar informação, e novamente o modo de armazenar essa informação iria precisar evoluir. As informações digitais passaram a ser mais comumente chamadas de

¹ <https://www.biblio.campusanandindeua.ufpa.br/index.php/ultimas-noticias/573-conheca-a-biblioteca-de-Alexandria>

dados, e os lugares em que elas são armazenadas são as bases de dados, e as ferramentas que usamos para gerenciá-las são os bancos de dados.

Existindo modos de gerar, armazenar e recuperar informação numa escala muito maior do que se tinha antes, passamos a enfrentar outro problema, o acesso a essa informação tende a ficar cada vez mais complexo e demorado. A encyclopédia ² define “Acesso a informação” como “A habilidade de identificar, recuperar, e usar a informação efetivamente”, ou seja, o acesso é considerado uma propriedade da informação, de modo que pode-se estabelecer que existe a necessidade de não apenas ter a informação guardada, mas de ter um meio de se obtê-la. Com isso, chegamos no tema deste trabalho que visa apresentar dois modos diferentes de armazenamento digital de informação, e quantificá-los de maneira que seja possível determinar, dentro de parâmetros específicos, o que tem o melhor desempenho.

Atualmente existem no mercado duas filosofias principais para gerenciamento de dados, o modelo relacional e o não relacional. O modelo relacional é o modelo clássico utilizado continuamente há mais de 50 anos para a finalidade de gerenciar informações, continuando a ser o mais popular para esse fim. Ele tem uma estrutura rígida, que necessita de um estudo complexo das informações sendo armazenadas, obtendo seu nome por conta de seus dados serem representados por relações, que são tabelas com linhas e colunas (MACÁRIO; BALDO, 2005). Já o modelo não relacional tem uma abordagem mais flexível ao gerenciar informações digitais que supostamente se adapta melhor aos diferentes cenários atuais com grandes quantidades de informações. O modelo não relacional promete lidar melhor, em especial com a escalabilidade da informação, que é a capacidade de manter o desempenho do gerenciamento dos dados mesmo quando a quantidade deles aumenta muito.

A inspiração desse trabalho veio de uma experiência pessoal que eu gostaria de narrar brevemente. No começo de 2023, precisei transferir os dados de uma ferramenta antiga da empresa em que trabalhava, para uma ferramenta nova que usava um método diferente de armazenamento de dados. Nessa tarefa, os dados existentes precisavam ser alterados para um formato que a tecnologia mais nova reconhecesse antes que pudessem ser armazenados, mantendo também as relações entre as informações existentes. Essas mudanças também precisavam ocorrer de maneira que não houvesse conflitos com a tecnologia nova. Mantendo o exemplo das bibliotecas como locais de armazenamento, minha tarefa seria algo como reescrever toda a informação de uma biblioteca que utilizava papiros, mas que gostaria de se modernizar para usar livros. A informação nesta biblioteca poderia ser a mesma, mas os livros são muito diferentes dos papiros, o que acarretaria na necessidade de várias mudanças para que a biblioteca pudesse suportar a nova tecnologia.

A experiência citada acima gerou em mim uma grande curiosidade sobre o gerenciamento de informações digitais, em especial sua performance em diferentes situações

² <https://www.encyclopedia.com>

com diferentes quantidades e tipos de informação. Durante o decorrer deste trabalho, espero adquirir o conhecimento necessário, por meio de experimentação e testes, para que possa dizer com segurança se a decisão tomada pela empresa foi a acertada, e responder a pergunta que ficou na mente durante todo esse tempo, Valeu a pena?

2 Revisão de Literatura

Neste capítulo será apresentada a revisão da leitura, mostrando definições das principais informações e características em artigos científicos que justificam a realização deste trabalho. Na seção 2.1 o tema de dados será abordado com mais profundidade com uma explicação dos tipos de dados pertinentes para esse artigo. A seção 2.2 será responsável por definir bancos e bases de dados, e conceitos importantes para compreendê-los. A seção 2.3 será responsável por detalhar as características de bancos de dados relacionais a seção 2.4 irá descrever os sistemas gerenciadores de bancos de dados não relacionais, contextualizar o leitor em sistemas distribuídos e detalhar suas diferenças e vantagens em comparação com o modelo relacional.

2.1 Dados

Como foi citado acima, apesar de não serem sinônimos, "dado" é um termo comumente usado para se referir a informações, e na área da informática são responsáveis por representar digitalmente informações do mundo real. Mais precisamente, dados são fatos que podem ser gravados e tem um significado implícito (ELMASRI et al., 2005). De modo geral, podemos classificar os dados em duas principais categorias:

- **Dados relacionais:** Assim como o modelo relacional já citado, eles derivam seu nome por representarem relações sendo propícios para serem armazenados em tabelas. A presença de um esquema de dados rígido impede que o formato das informações seja livremente alterado, aumentando o peso das decisões iniciais ao usar esse tipo de dado. Esses fatores contribuem para uma alta integridade, porém dificulta o escalabilidade (GARCIA; SOTTO, 2019).
- **Dados não relacionais:** São dados que de algum modo fogem do que se pode armazenar facilmente em tabelas relacionais como imagens, coordenadas e dados gerados por sensores. A ausência de um esquema, ou a implementação de um esquema mutável permite grande flexibilidade dos dados armazenados e facilita sua disponibilidade, tornando implementações com dados não relacionais mais escaláveis (GARCIA; SOTTO, 2019).

2.2 Bancos de Dados

De um modo geral, bancos de dados são simples, eles guardam informações, porém é interessante usar a comparação de Date e pensar como em bancos de dados como armários

eletrônicos de armazenamento (DATE, 2004). Nessa analogia, uma pessoa organizada fará com que cada gaveta do armário seja responsável por um tipo de informação, por exemplo uma gaveta para remédios, uma para dados do carro, uma para documentos pessoais. Idealmente o armário tem uma ideia geral que a informação deve respeitar para ser guardada dentro dele, não faz muito sentido guardar os itens de cozinha no armário de documentos (mas claro que cada pessoa decide sua necessidade).

Seguindo essa linha de pensamento podemos ver que os bancos de dados tem a mesma função que os armários, eles armazenam informações semelhantes que têm algum tipo de relacionamento entre si. Essa também é a conclusão de Korth e Berg que dizem, respectivamente, que bancos de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico (KORTH HF, 1994) e uma coleção auto-descrita de registros integrados (BERG et al., 2013).

Ficando clara a definição de um banco de dados, fica mais fácil de entender sua função como sistemas de armazenamento de dados. Eles trabalham informações digitais de modo que é formado um esqueleto para como elas serão guardadas e posteriormente usadas. Em um mundo cada vez mais digital (COFFMAN; ODLYZKO, 1998), muito mais informações são geradas e necessitam de um lugar para serem armazenadas, tornando bancos de dados ferramentas fundamentais.

Relacionados aos Bancos de Dados existem também os Sistemas gerenciadores de Bancos de Dados, ou SGBD's, que é a junção do banco de dados com uma série de ferramentas que auxiliam no acesso às suas informações (PART, 2006).

Com o tempo, tanto os bancos de dados quanto os dados em si foram evoluindo e tomando formas diferentes do que foi originalmente concebido. Quantidades massivas de dados e diferentes necessidades de armazenamento forçaram a infraestrutura digital de tal modo que novos conceitos e ideias precisaram ser concebidas. Hoje, há uma grande gama de diferentes sistemas gerenciadores de bancos de dados, cada um com suas vantagens e especialidades, e de modo geral podem ser divididos em duas categorias:

2.3 Bancos Relacionais

Um Banco de dados relacional é um banco de dados percebido por seus usuários como uma coleção de RelVars (relvars) ou, de modo mais informal, tabelas (DATE, 2004). Eles aplicam os conceitos de álgebra relacional e tem uma estrutura formada por tabelas (ALVES, 2014), que são formadas por colunas e linhas, onde essas linhas devem ser um registro único no banco evitando a redundância das informações armazenadas. As tabelas de um banco relacional são interligadas de maneiras naturais chamadas de relacionamentos (PAREDAENS et al., 2012), que se manifestam com um valor de uma linha sendo uma referência direta ao valor de outra linha em outra tabela. Essa ligação entre as informações que dá o nome do modelo, as informações se relacionam entre si.

De modo geral, os bancos de dados relacionais são projetados para manter a consistência dos dados, sacrificando a facilidade com que grandes quantidades de dados podem ser processadas (GARCIA; SOTTO, 2019).

Nas seções seguintes serão detalhados conceitos importantes para o tema de bancos de dados relacionais, algumas propriedades importantes para a compreensão do modelo relacional serão listadas e descritas com uma breve explicação de cada propriedade e sua função na implementação de bases relacionais. Em seguida o papel da linguagem SQL será apresentado no contexto do modelo relacional.

2.3.1 ACID

Os bancos relacionais têm um grande foco na consistência dos dados, sendo composto por tabelas onde as linhas representam um conjunto único de informações no banco. Com o constante fluxo de dados, a tarefa de mantê-los consistentes com o resultado esperado após inúmeras transações (uma operação tratada como uma unidade de trabalho, atômica e sem estado intermediário ¹) é muito complexa. Com isso em mente, podemos imaginar as propriedades ACID como uma série de regras que permite que o banco de dados atue de forma consistente, e que as informações guardadas nele sejam confiáveis. Descrevendo o acrônimo, temos:

- **Atomicidade:** Consiste no mecanismo que faz com que ou uma transação seja executada em sua integridade, ou ela não será executada.
- **Consistência:** Caso haja alguma falha em uma transação, os dados devem retornar ao seu estado inicial pré transação.
- **Isolamento:** Garante que uma transação que está ocorrendo não pode ter interferência por outras transações até que tenha sido completada.
- **Durabilidade:** Mantém os dados disponíveis e em seu estado correto mesmo após uma falha/ reinício do sistema.

2.3.2 SQL

O SQL ou Structured Query Language (Linguagem de busca estruturada) é uma linguagem de busca usada para acessar o conteúdo de bases de dados relacionais. Inicialmente implementada pela IBM, foi adotada em massa a partir dos anos 70 para a construção de vários bancos de dados comerciais (como Oracle e DB2). É uma linguagem muito influente e difundida para uso em bases de dados relacionais, sendo uma das mais importantes fundações da tecnologia da indústria da computação hoje em dia e emergindo

¹ <https://www.databricks.com/br/glossary/acid-transactions>

como a linguagem padrão para uso em bases de dados relacionais (GROFF; WEINBERG; OPPEL, 2002):

Em relação aos bancos de dados relacionais, o SQL permite:

- Definição de dados, onde pode-se definir a estrutura e organização assim como os relacionamentos entre seus dados
- Recuperação de dados, que é o que permite a leitura das informações persistidas na base
- Manipulação de dados, que são as operações de inserção, remoção e alteração de dados no banco;
- Controle de acesso, onde pode-se restringir o acesso e permissão de usuários as funções do sistema
- Compartilhamento de dados, que coordena o acesso concorrente de usuários
- Integridade de dados, define restrições de integridade, protegendo os dados de corrompimento

Dito isso, a linguagem é considerada não completa pois não permite teste de condições ou comandos para fluxo de controle, sendo especializada em em tarefas de gerenciamento de banco de dados.

2.4 Bancos não relacionais

Apesar das suas qualidades, sendo uma tecnologia antiga do ponto de vista da informática, implementados inicialmente no início da década de 70 (CODD, 1970), os bancos de dados relacionais foram com o tempo apresentando desempenhos insatisfatórios em certas áreas, notavelmente em relação a escalabilidade (GROFF; WEINBERG; OPPEL, 2002) e armazenamento de dados não relacionais. Essa insatisfação gerou espaço para que outras alternativas fossem consideradas, resultando na popularização de bases não relacionais.

Os bancos de dados não relacionais, ou NoSQL (Not Only SQL), propõem atender melhor aos requisitos atuais de armazenamento de dados, que incluem o armazenamento de informações que não seguem o modelo relacional (como documentos e dados de GPS), o armazenamento e gerenciamento de quantidades massivas de dados (LÓSCIO; OLIVEIRA; PONTES, 2011), e em especial a necessidade de um banco altamente escalável. Para que esses objetivos sejam atingidos, vários bancos NoSQL fazem a separação do banco de dados em nós, ou em instâncias, desse banco que podem ser gerenciadas em recursos distribuídos de maneira horizontal, sem dependência entre eles. Também são aplicadas técnicas avançadas como o Sharding O que é o processo de separação do banco de dados

em bancos menores, ele é notoriamente difícil de ser feito em bancos relacionais pois quebra a cadeia de relacionamentos, porém isso não é um fator nos bancos não relacionais possibilitando sua implementação de maneira mais eficiente ²

Essas características tornam os bancos não relacionais muito atrativos para solucionar vários dos problemas que os ambientes virtuais enfrentam hoje. Como não dependem de esquemas, existe uma flexibilidade grande no tipo de dado que pode ser armazenado nos bancos, sua separação em nós permite que a infraestrutura cresça de maneira sustentável e de maneira horizontal garantindo a disponibilidade das informações. Os bancos de dados relacionais ainda predominam no mercado, porém existe um crescimento notável no uso de bancos de dados não relacionais (GYORÖDI; GYORÖDI; SOTOC, 2015), provando que o modelo tem sua relevância no mercado.

Como consequência dessas características, os bancos não relacionais não poderiam implementar as propriedades ACID como os bancos relacionais, pois não são adequadas a esse modelo. Porém os bancos não relacionais são no fundo sistemas distribuídos, fazendo com que devam seguir as regras do teorema CAP, será detalhado a seguir.

2.4.1 Sistemas distribuídos e CAP

Sendo que a capacidade de crescimento sustentável é uma das principais características atrativas nos bancos de dados não relacionais, é importante que o conceito de sistemas distribuídos esteja bem claro. Os sistemas distribuídos têm a capacidade de manter os recursos que estão fisicamente distantes, disponíveis para os consumidores do sistema, por exemplo uma grande empresa com várias plantas (KLEINROCK, 1985) pode usar essa arquitetura para manter consistência nas informações entre as plantas e a matriz. Essa característica é interessante por dois principais motivos:

- Escalabilidade: Os recursos necessários para o sistema podem ser adquiridos conforme a necessidade da aplicação, assim não é necessário desperdiçar recursos criando uma estrutura robusta demais para uma aplicação que está no início da sua vida. E caso o uso do sistema supere as expectativas iniciais, mais nós podem ser alocados para suprir a demanda sem que a estrutura do sistema seja alterada de modo significativo.
- Disponibilidade: Como a estrutura está distribuída ao redor da rede, o sistema está mas protegido de situações que fariam que um sistema monolito ficasse completamente indisponível, pois é possível fazer com que as responsabilidades de um nó do sistema distribuídos sejam passadas para outro caso haja um problema.

Os sistemas distribuídos resolvem muitos dos problemas de sistemas modernos, porém eles ainda tem suas limitações e devem seguir alguns princípios para manter a confiabilidade

² <https://www.devmedia.com.br/banco-de-dados-nosql-um-novo-paradigma-revista-sql-magazine-102/25918>

dos serviços que eles prestam. Esses princípios são detalhados pelo teorema CAP que foi apresentado inicialmente na PODC em 2000. Introduzido para detalhar uma característica importante de sistemas distribuídos, que eles necessariamente devem fazer uma troca entre sua consistência (consistency), disponibilidade (availability) e tolerância a partição (Partition tolerance) (GILBERT; LYNCH, 2012) sendo que cada um desses aspectos significa:

- **Consistência:** Garante que o consumidor dos dados do banco, terá acesso aos mesmos dados independente de qual nó do banco estará acessando.
- **Disponibilidade:** Operações feitas em um servidor que por algum motivo não está disponível, serão redirecionadas para outros servidores que podem dar sequência ao seu processamento.
- **Tolerância ao particionamento:** Dita que mesmo que a comunicação entre as bases de dados do sistema seja quebrada, o sistema ainda deve estar funcionando (LEE et al., 2021)

O teorema CAP dita que é impossível num sistema distribuído implementar essas três características simultaneamente (a prova pode ser vista em (GILBERT; LYNCH, 2012)) , e garante que duas das três constituintes (CAP) do teorema podem ser entregues a qualquer momento, com as constituintes combinadas sendo CP (consistência e resistência a partição), AP (disponibilidade e resistência a partição) ou CA (consistência e disponibilidade) (TATSIS, 2022).

3 Descrição conceitual

Esse capítulo descreve a ideia geral da implementação do trabalho, e detalhes importantes a serem mencionados. Na seção 3.1 será detalhado o algoritmo usado no meu trabalho para fazer a conversão de dados relacionais para não relacionais, e como eu pretendo reproduzir isso neste trabalho, seção 3.2 é responsável por detalhar os SGBDs usados para a pesquisa. A seção 3.3 apresenta o algoritmo usado para a execução das operações nos bancos de dados, assim como a preparação dos dados para serem manipulados e o hardware e software onde o algoritmo foi executado. A seção 3.4 é responsável por detalhar quais operações foram efetuadas nos bancos de dados, o por que que essa procura é pertinente para o texto e como ela poderá gerar resultados conclusivos. Na 3.5 os métodos de avaliação dos resultados é discutido e por fim na seção 3.6 as expectativas para os resultados são comentadas.

3.1 SGBDs

Como já foi comentado, a inspiração desse trabalho veio de uma atividade que eu tive que fazer em um trabalho antigo. Nesse trabalho, eu precisei fazer a conversão de uma base relacional que usava PHPMyAdmin para uma base nova em MongoDB. Para fazer essa conversão, todos os dados da base relacional foram exportados para JSON, sendo cada tabela um arquivo. Com esses dados exportados, criei um script em node para cadastrar os dados, que pegava as informações dos arquivos JSON e as cadastrou hierarquicamente em documentos. Inicialmente eram cadastradas as tabelas que não tinham chaves estrangeiras, e em seguida as tabelas que tinham chaves estrangeiras eram cadastradas de modo que uma tabela só poderia ser cadastrada caso a tabela à qual ela referenciava já tivesse sido cadastrada. assim pude reproduzir os relacionamentos do banco relacional no banco não relacional, fazendo com que ao invés de existir uma coluna de chave estrangeira como no banco relacional, existisse no documento do banco não relacional uma chave cujo valor apontava para o registro de um outro documento. Para esse trabalho, foi feito um processo muito parecido, porém os dados foram inicialmente criados com JSON e não precisaram ser exportados, mas a ideia de manter os relacionamentos com chaves e valores que apontavam para os documentos corretos se manteve a mesma. Assim, neste trabalho, assim como foi necessário fazer na minha atividade, temos duas bases com os relacionamentos idênticos, porém diferem quanto a filosofia de tratamento desses dados.

3.2 SGBDs

Existem vários exemplos de sistemas gerenciadores de bancos de dados que aplicam o modelo relacional, como o MySQL, MariaDB, Oracle e PostgreSQL. Entre estes, o escolhido para a implementação do trabalho proposto nesse texto foi o PostgreSQL, por ser o banco em que eu tenho mais experiência e também trazer um plano gratuito aceitável, permitindo que a reprodução dos resultados desse texto seja feita sem acarretar em custos financeiros. Uma de suas características são suas licenças para uso gratuito, seja para fins estudantis seja para realização de negócios, possibilitando que empresas o usem livremente (MILANI, 2008).

Para a manipulação e gerenciamento do banco de dados relacional será usado o SGBD PostgreSQL na versão 16.1-1 e o administrador PgAdmin 4 na sua versão 6.19 para administrar o banco e gerar o diagrama de entidade relacional.

No modelo não relacional existe uma gama grande de SGBDs que aplicam uma variedade de métodos de armazenamento não relacional, O AzureCosmosDB por exemplo armazena os valores numa tabela hash, já bancos como Apache Giraph usam estruturas de grafos. Para esse texto o banco escolhido foi o MongoDB, pois foi o banco não relacional para o qual a empresa que eu trabalhava resolveu mudar sua estrutura previamente relacional, sendo assim um banco que tenho experiência e foi o responsável por inicialmente gerar a dúvida que deu origem a esse texto. O MongoDB é um Sistema gerenciador de banco de dados que aplica as propriedades CP (consistência e resistência a partição) do teorema CAP. Projetado para desenvolver rapidamente aplicações web e infraestrutura de internet (HOWS; MEMBREY; PLUGGE, 2019) ele tem várias facilidades em relação a sua implementação e disponibilização na rede. Entre os bancos de dados não relacionais ele se classifica como um banco orientado a documentos, o que significa que ao invés de armazenar dados em tabelas com colunas e linhas, ele armazena documentos no formato JSON ¹ permitindo uma grande flexibilidade nas informações armazenadas. Assim como o PostgreSQL, o MongoDB possui uma versão gratuita com uso muito amigável, permitindo a reprodução dos dados gerados nessa pesquisa com facilidade e sem custos.

A versão usada para essa pesquisa é o MongoDB Community Server 7.0.4 que permite a criação de uma instância local do banco de dados.

3.3 Algoritmo, software e hardware

Inicialmente, serão gerados dados com uma ferramenta geradora de dados sintéticos, a Mokaroo ², esses dados serão usados inicialmente para gerar uma base de dados semelhante a algo que existiria em um ambiente de produção real, aumentando a validade

¹ o JSON ou JavaScript Object Notation ou Notação de Objetos do JavaScript é um tipo formatação leve de dados altamente empregado na WEB para troca de informações de sistemas computacionais

² <https://www.mockaroo.com/>

e confiabilidade dos resultados. Esses dados serão gerados no formato JSON e serão consumidos pela aplicação para inicialmente armazenar os dados nos bancos de dados.

Após a geração dos dados, inicialmente serão aplicadas algumas das operações mais comuns que podem ser executadas em um bancos de dados para gerenciamento de dados, as operações CRUD (Create, Read, Update, Delete, ou Criar, Ler, Atualizar e Deletar). Os dados estarão contidos num arquivo que será lido e as informações serão devidamente cadastradas e os relacionamentos serão montados. Todo esse processo será registrado e o tempo de cada inclusão será armazenado para que ao final das operações possa-se calcular uma média de tempo. Após a criação, as operações de atualização, leitura e remoção também serão efetuadas do mesmo modo e com as mesmas métricas, visando geração de resultados consistentes.

As operações detalhadas acima serão executadas por um script simples implementado em javascript com o motor Node na versão 19.7.0. O objetivo desse algoritmo é apenas fazer a conexão com os banco de dados e chamar os métodos necessários das instâncias do banco de moto automático, permitindo que uma quantidade grande de dados seja manipulada ao mesmo tempo.

Por fim, todo esse trabalho será executado na mesma máquina, com a intenção de manter os resultados os mais restritos ao desempenho dos bancos apenas. O computador em que serão efetuados os testes é um ThinkPad Gen2 com processador AMD Ryzen 7 PRO 5850U com Radeon Graphics e 1.90 GHz de frequência base, 16Gb de memória RAM 3200 GHz e um SSD m.2.

3.4 Procuras

Uma série de pesquisas de complexidade incremental foram executadas para avaliar o desempenho dos sistemas de bancos de dados. O resultado dessas procuras é de suma importância pois dá uma forte indicativo a eficácia do sistema gerenciador de banco de dados, e os resultados dessa pesquisa podem ser importantes para fruta referência em relação a decisões de tecnologia. As pesquisas executadas foram implementadas para testar vários aspectos dos bancos de dados, desde como eles se comportam executando pesquisas mais simples porém muito utilizadas no dia a dia de um sistema, até pesquisas criadas para relacionar o maior quantidade de dados possíveis. As pesquisas resultantes dessas considerações foram:

1. Por ID: A procura por ID consiste no retorno de uma linha de tabela (no caso da base relacional) ou de um documento (no caso do MongoDB) quando se tem um identificador único para essa informação. Apesar de uma procura simples, é muito utilizada em ambientes de produção, o resultado de sua performance trará um dado importante para a pesquisa.

2. Listagem geral: Aqui, serão listados todos os dados de uma determinada tabela, essa procura pode não ter uma complexidade grande, mas gera uma quantidade grande de dados de resposta e a análise de sua performance pode gerar uma boa ideia de como o banco de dados reage a quantidades grandes de informações geradas nas buscas
3. 1:N O relacionamento 1:N ou um para muitos representa situação onde um registro de um tipo pode ter um relacionamento com um ou muitos registros de outros tipos, mas esses registros têm apenas o primeiro registro no seu relacionamento. Um clássico exemplo seria o de uma pessoa que pode ter vários carros, mas o carro em si pode ter apenas um dono. No modelo proposto as operações que serão efetuadas serão, listagem do paciente com seus exames e com as datas de descoberta das suas doenças, doenças, e médico com os exames que executou
4. N:N O relacionamento N:N ou muitos para muitos é o relacionamento onde um objeto de um tipo pode se relacionar com um ou vários objetos de outros tipos, e esses por duas vezes podem também se relacionar com um ou vários objetos do primeiro tipo. Essa é a pesquisa mais interessante para esse estudo, pois é a mais complexa e trabalhosa tanto de se escrever como de se executar, porém ainda é sim muito presente no dia a dia de quem precisa usar bancos de dados para recuperar informações. Aqui vai-se realmente comparar a capacidade dos bancos de lidar com uma quantidade grande de informações, e também com uma complexidade grande no relacionamento das tabelas e documentos. Serão executadas as operações de busca de quais médicos atendem quais pacientes, e também quais médicos atendem quais doenças

3.5 Métodos de avaliação

Todas as operações serão efetuadas 1000 vezes, fazendo a média calculada do desempenho dos testes seja o mais confiável possível. As bases também serão populadas com uma quantidade grande de informações, na tentativa de simular uma base de dados real, que já tem algum tempo de uso em ambientes com populares. O principal método de avaliação será o tempo, pois é ele que mais importa para o usuário final. O tempo será fornecido pelos testes e informações como a média, mediana, desvio padrão e variância serão processados e apresentados, porém o dado principal para a retirada das conclusões será a média pois, reiterando, o foco desse texto é ajudar na tomada de decisão em relação a que tecnologia de armazenagem de dados se usar, logo caso haja algo no SGBD que gera uma inconsistência que afete os resultados, isso será tomado como um problema da ferramenta.

3.6 Expectativas

No levantamento bibliográfico para esse texto, viu-se que a performance das bases de dados NoSql foi muito elogiada, em especial com grandes quantidades de dados, o que era de se esperar visto que uma das principais características do modelo não relacional é a escalabilidade. Além disso existe também o fator da minha experiência pessoal que foi bastante positiva, com o MongoDB entregando resultados muito satisfatórios em pesquisas complexas. Esses fatores contribuíram para que a expectativa inicial fosse de que os bancos não relacionais tivessem um desempenho superior ao modelo relacional mais antigo. Apesar dessa expectativa positiva, deve-se considerar que a quantidade de dados usada nos testes pode não ser grande o suficiente para que essas vantagens dos bancos não relacionais aflorem, mantendo ainda a possibilidade do modelo relacional se manter relevante no contexto desse texto. Uma sugestão de trabalho futuro seria a procura do tamanho de banco de dados mínimo no qual o modelo no sql passe a ser mais efetivo.

4 Resultados e Discussão

Nesta seção, serão apresentados os resultados obtidos durante a pesquisa. Também será detalhado o método que foi empregado para se obter os resultados apresentados, assim como quaisquer pontos importantes a serem mencionados. O Modelo de dados e detalhes dos bancos de dados serão usados na pesquisa serão detalhados na seção 4.1. Na seção 4.2 haverá detalhes dos dados sintéticos que foram gerados para simular a base de dados. A seção 4.3 ficou responsável por explicar com mais detalhes o processamento dos dados e operações nos bancos de dados. Os testes efetuados na pesquisa serão detalhados na seção 4.4 e os resultados serão apresentados na seção 4.5

4.1 O Modelo

O primeiro passo foi estabelecer o que seria armazenado na base de dados. A estrutura criada foi projetada para fornecer os desafios corretos aos bancos de dados, era necessário que houvesse pelo menos um relacionamento 1:N e um N:N para que as procuras mais complexas pudessem ser efetuadas. Assim, foi projetada uma base simulando uma clínica médica, que contém Pacientes, Médicos, Doenças e Exames, sendo que um exame tem um médico e um paciente, e as doenças podem estar presentes em vários pacientes assim como pacientes podem ter várias doenças. Desse modo, foi criado o seguinte modelo relacional:

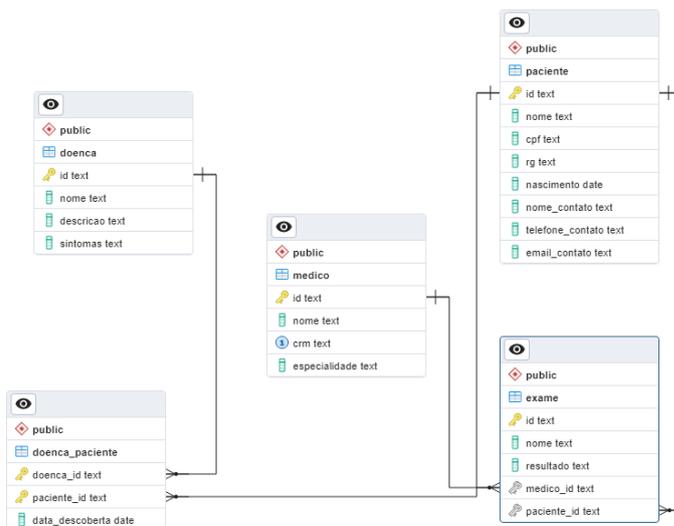


Figura 1 – Diagrama de Modelo Relacional

4.2 Dados sintéticos

Como já foi citado, os dados foram gerados a partir de uma ferramenta chamada mokaroon. Nesta ferramenta, pode-se criar um modelo da base de dados, e atribuir um tipo de valor a ser gerado para aquele campo,. Assim, foram criados dados para pacientes, médicos, doenças, exames e doenças de pacientes, que puderam ser exportados para arquivos JSON. Ao todo, foram criados 1000 pacientes, exames, doenças, e doenças de pacientes, e 100 médicos. Essa proporção foi escolhida para fazer mais sentido no contexto de uma clínica, que vai ter menos médicos do que pacientes.

4.3 Processamento de dados e operações

O algoritmo para inserção e procura foi implementado de maneira simples, os dados gerados na ferramenta mokaroon foram armazenados em arquivos JSON que puderam ser importados pelo script. Com os dados importados, manipulá-los virou uma questão de iterar o vetor de informações e criar funções para fazer as operações necessárias. Para cada operação, foi criada uma função que fazia uma operação específica, ao todo, foram implementadas 7 funções que compreendiam criar, ler, alterar, deletar, listar todos, executar uma pesquisa 1:N e executar uma pesquisa N:N , isso para cada banco de dados, totalizando 14 funções diferentes. Essas funções imprimiam o tempo em milissegundos de execução da tarefa específica em um arquivo, para que os dados pudessem ser coletados. Ao todo, foram gerados 14 arquivos com 4100 linhas de dados coletados cada uma.

Após a geração dos resultados, cada arquivo gerado foi exportado para uma coluna em um arquivo CSV representando qual operação que ele executou. Então, usando a ferramenta de planilhas da Google foram calculadas a média, mediana, desvio padrão e variação de cada coluna.

4.4 Testes efetuados

Inicialmente nos testes , as operações CRUD foram feitas nos bancos. As operações CRUD, ou Create, Read, Update e Delete (Criar, Ler, Atualizar e Deletar) são algumas das operações mais básicas no dia a dia de um banco de dados e seu desempenho é de grande importância para um SGBD. Para executar essas operações, os arquivos JSON foram importados, adicionados num vetor e iterativamente criados, alterados, lidos e deletados. Em seguida, entramos no âmbito das buscas mais completas que não só lidam com complexidade mas com quantidade de dados. Na implementação da busca em um relacionamento 1:N foram executadas 3 procuras, a procura de exames por paciente, de datas de descoberta de doença por paciente, e de exames por médico.

	A	B	C	D	E	F	G
1	CREATE	READ	UPDATE	DELETE	ALL	1:N	N:N
2	68	31	41	27	80	66	27
3	3	3	4	2	36	37	3
4	2	1	2	2	30	26	2
5	2	2	2	1	25	20	2
6	2	1	2	1	21	20	3
7	2	1	2	1	21	20	2
8	2	2	2	1	19	20	3
9	1	1	1	0	18	21	1
10	2	1	2	1	15	18	2
11	1	2	1	1	16	19	1
12	1	1	2	1	14	19	2
13	2	1	1	1	15	19	2
14	1	1	1	1	14	20	1
15	1	1	2	1	16	20	2
16	2	1	1	1	17	21	2
17	2	1	1	1	17	19	1
18	2	1	2	1	15	18	1
19	1	1	1	1	16	17	2
20	2	1	1	1	15	19	2
21	1	1	1	0	15	19	1
22	2	1	1	1	15	18	2
23	1	1	1	1	14	19	1
24	1	1	1	0	16	18	1

Figura 2 – Dados coletados na ferramenta Planilhas do google

Para os relacionamentos N:N, foram executados testes procurando quais pacientes foram atendidos por quais médicos (necessitando passar pela tabela de exames) e também procurando quais médicos atendiam quais doenças, necessitando que as tabelas médico, exame, paciente, doenca_paciente e doenca fossem incorporadas na procura. Essa procura em específico é de extrema importância para esse artigo, pois é a procura mais complexa, interagindo com a maior parte das tabelas ou documentos dos bancos de dados, e seu resultado pode dar indicações fortes do desempenho dos bancos em ambientes de produção.

4.5 Resultados obtidos

Após o tratamento dos dados, os valores que foram obtidos foram os seguintes para o modelo não relacional:

	Média	Média	Mediana	Desv. Pad	Varição
CREATE	0,9102439024		1	1,171442933	1,372278545
READ	0,592195122		1	0,6925791158	0,4796658316
UPDATE	0,8570731707		1	0,8220914305	0,6758343201
DELETE	0,602195122		1	0,6579000362	0,4328324576
TODOS	4,10756098		13	4,955364702	24,55563933
1:N	11,89404297		10	11,22186091	125,9301623
N:N	2,123902439		1	2,604585142	6,783863762

Tabela 1 – Tabela de resultados SGBD MongoDB (milissegundos)

Os gráficos a seguir mostram o resultado de cada operação de modo detalhado, sendo o eixo X as operações e o eixo Y o tempo em milissegundos:

Não relacional: CREATE

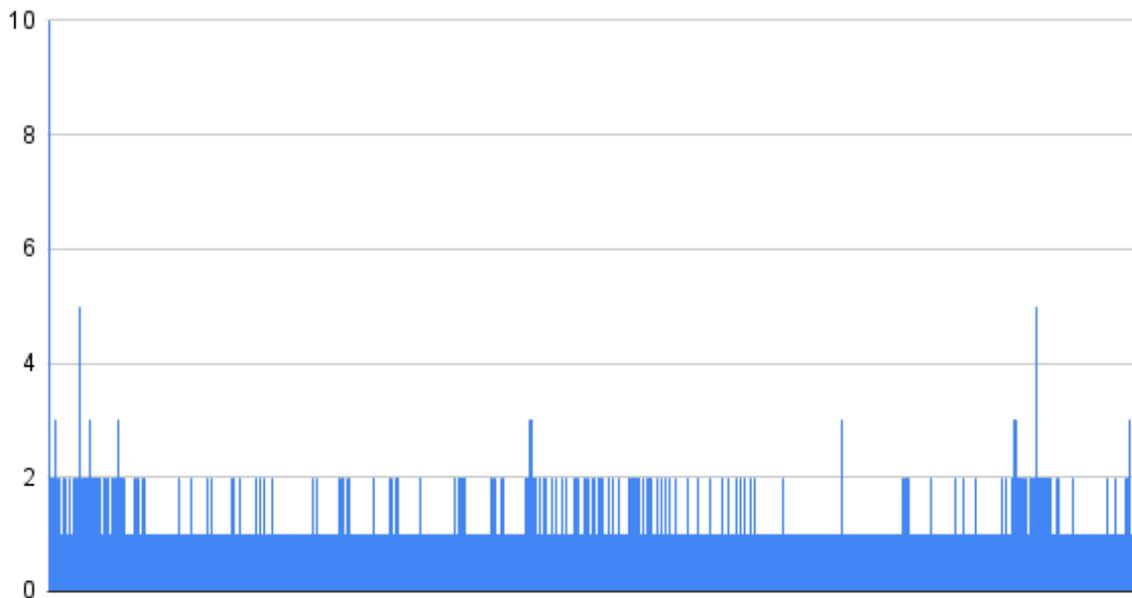


Figura 3 – Gráfico de tempo de execução de operações CREATE no SGBD MongoDB.

Não relacional: READ

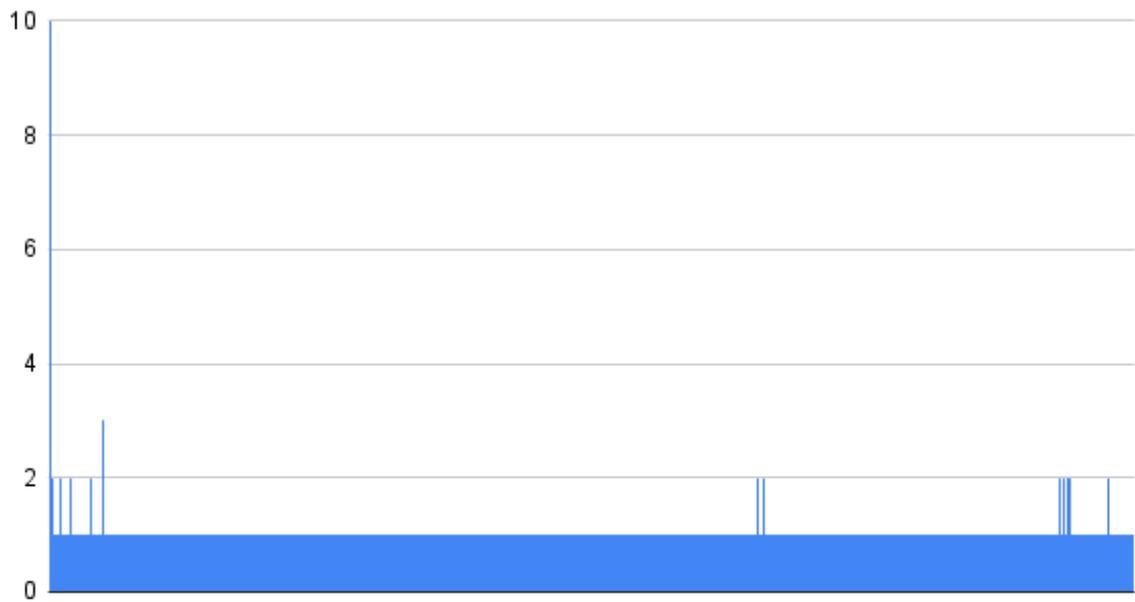


Figura 4 – Gráfico de tempo de execução de operações READ no SGBD MongoDB.

Não relacional: UPDATE

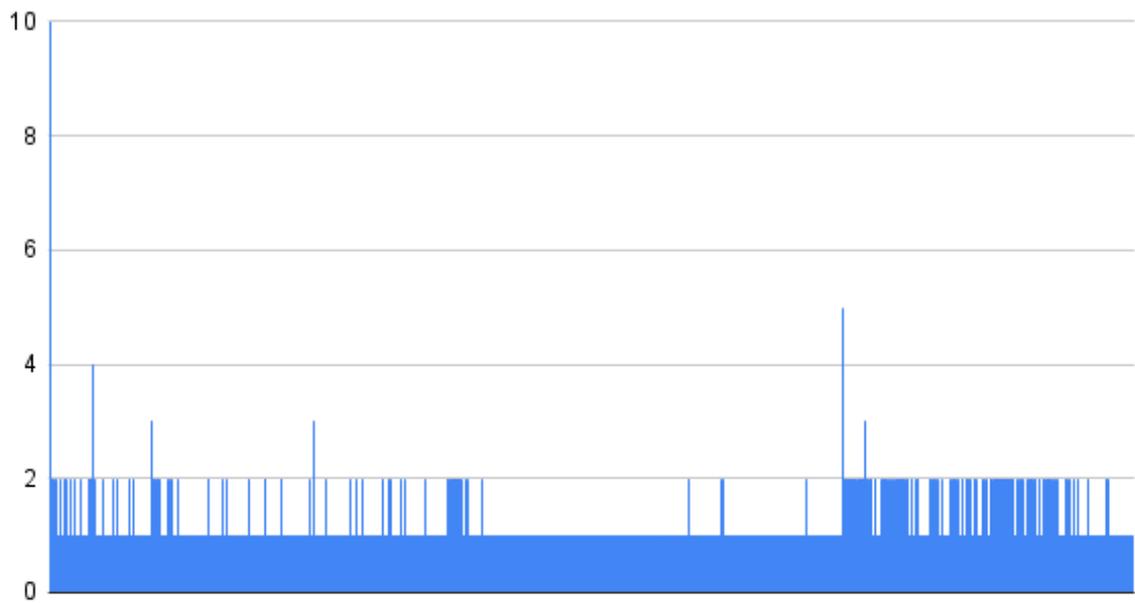


Figura 5 – Gráfico de tempo de execução de operações UPDATE no SGBD MongoDB.

Não relacional: DELETE

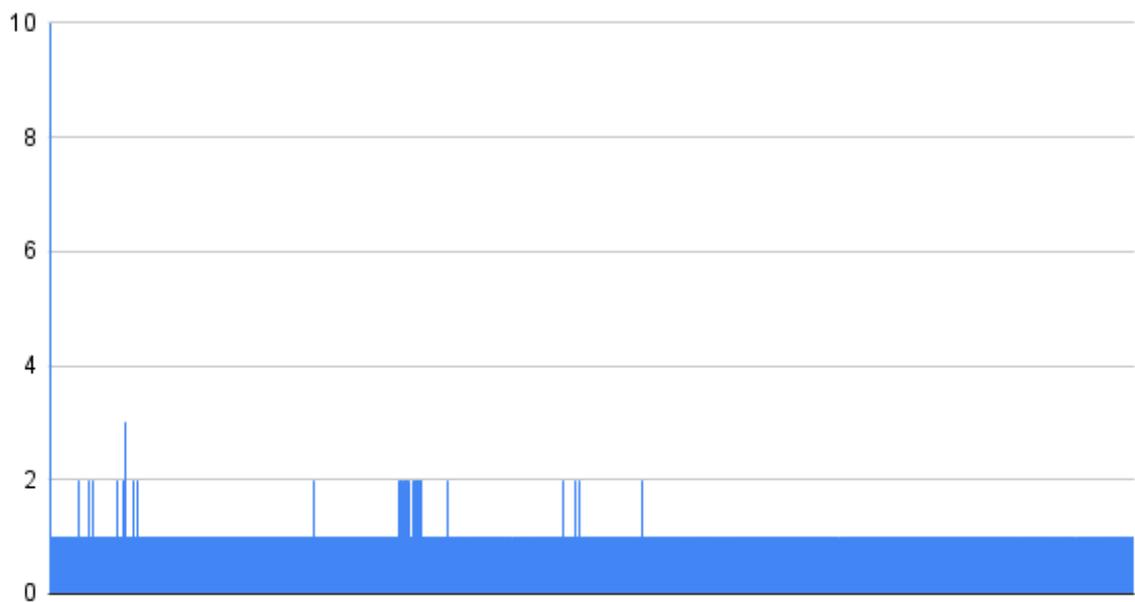


Figura 6 – Gráfico de tempo de execução de operações DELETE no SGBD MongoDB.

Não relacional: Todos

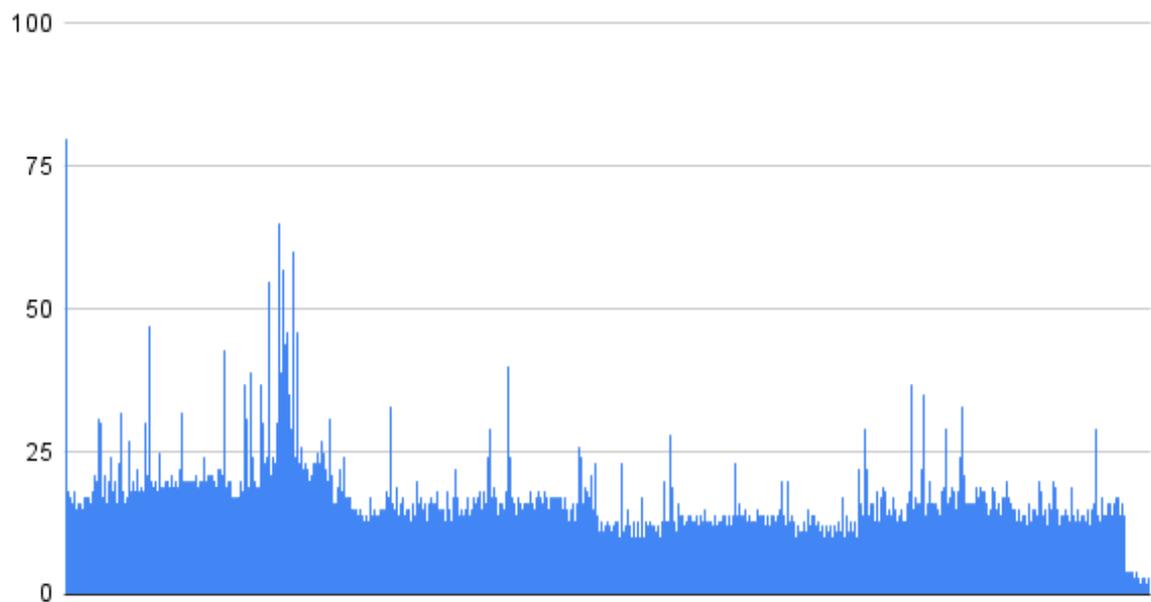


Figura 7 – Gráfico de tempo de execução de operações recuperando Todos os dados no SGBD MongoDB.

Não relacional: 1:N

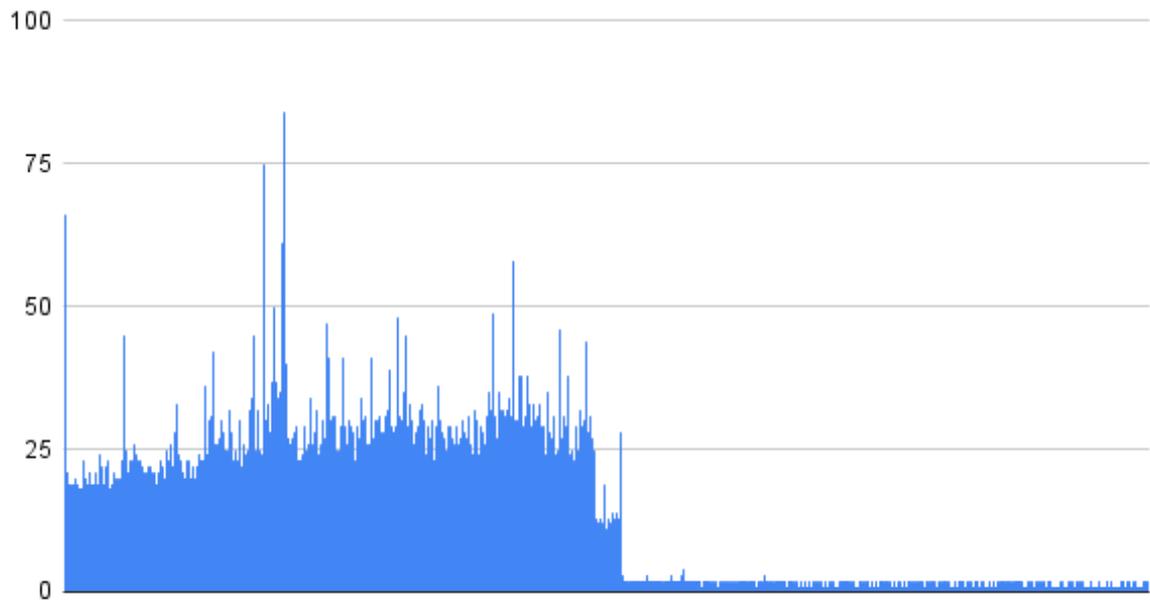


Figura 8 – Gráfico de tempo de execução de operações 1:N no SGBD MongoDB.

Não relacional: N:N

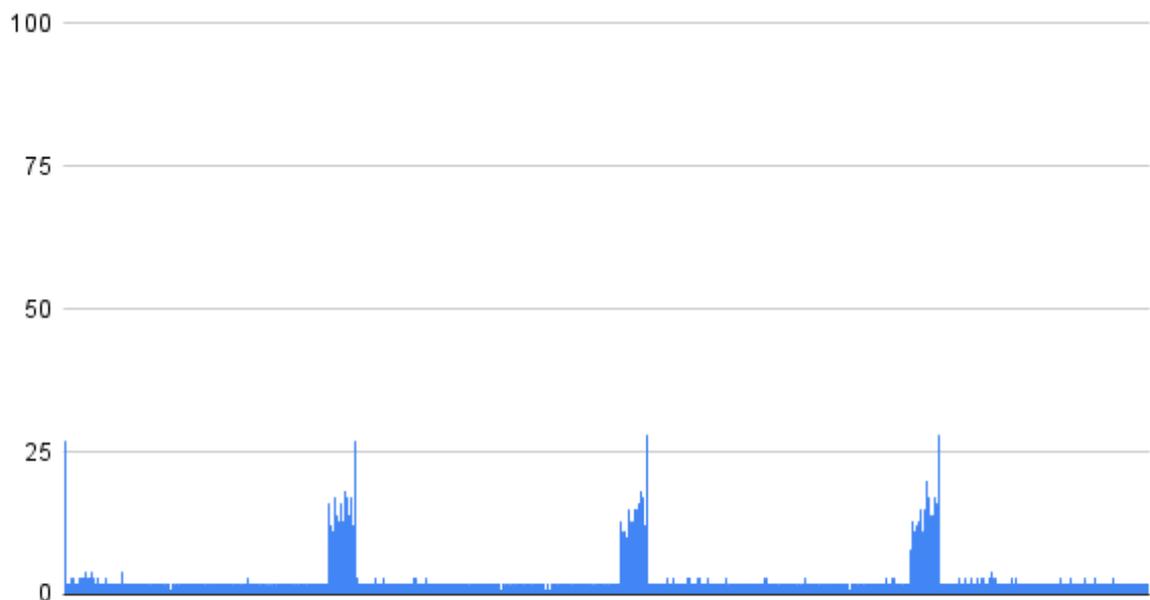


Figura 9 – Gráfico de tempo de execução de operações N:N no SGBD MongoDB.

Os mesmos testes foram aplicados para o modelo relacional, com os seguintes resultados:

	Média	Mediana	Desv. Pad	Varição
CREATE	0,3156097561	0	0,800951035	0,6415225605
READ	0,3804878049	0	2,278726196	5,192593077
UPDATE	0,4358536585	0	0,7784812521	0,6060330598
DELETE	0,5347645767	1	0,9393252264	0,8823318809
TODOS	0,3834146341	0	0,7522061876	0,5658141486
1:N	0,6348780488	1	1,14014466	1,299929846
N:N	0,9907317073	1	1,547148555	2,393668652

Tabela 2 – Tabela de resultados SGBD PostgreSql (milissegundos)

Relacional: CREATE

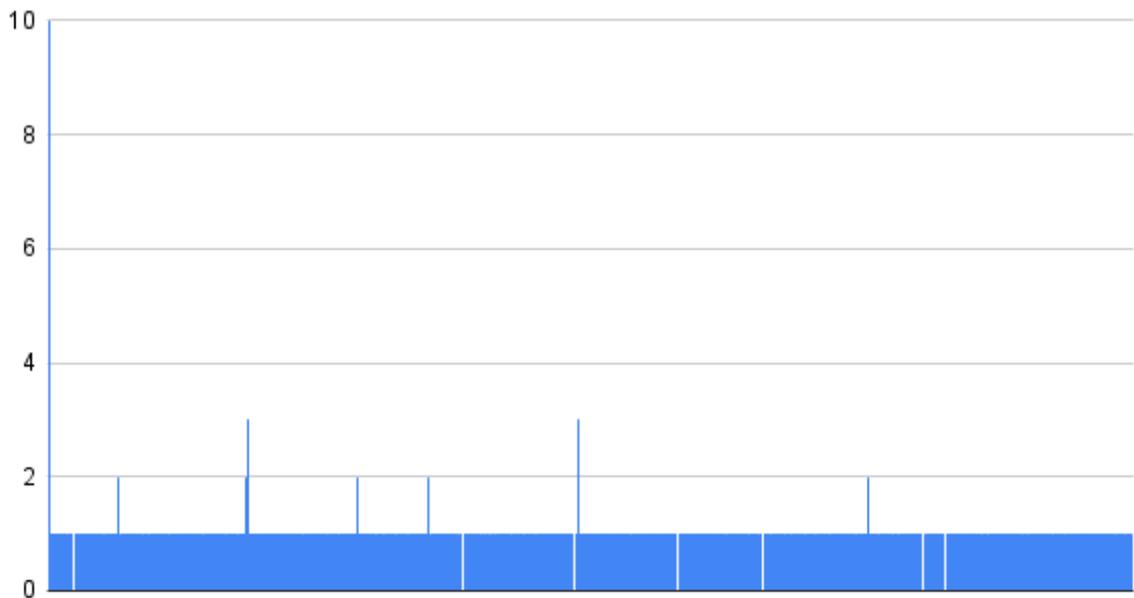


Figura 10 – Gráfico de tempo de execução de operações CREATE no SGBD PostgreSQL.

Relacional : READ

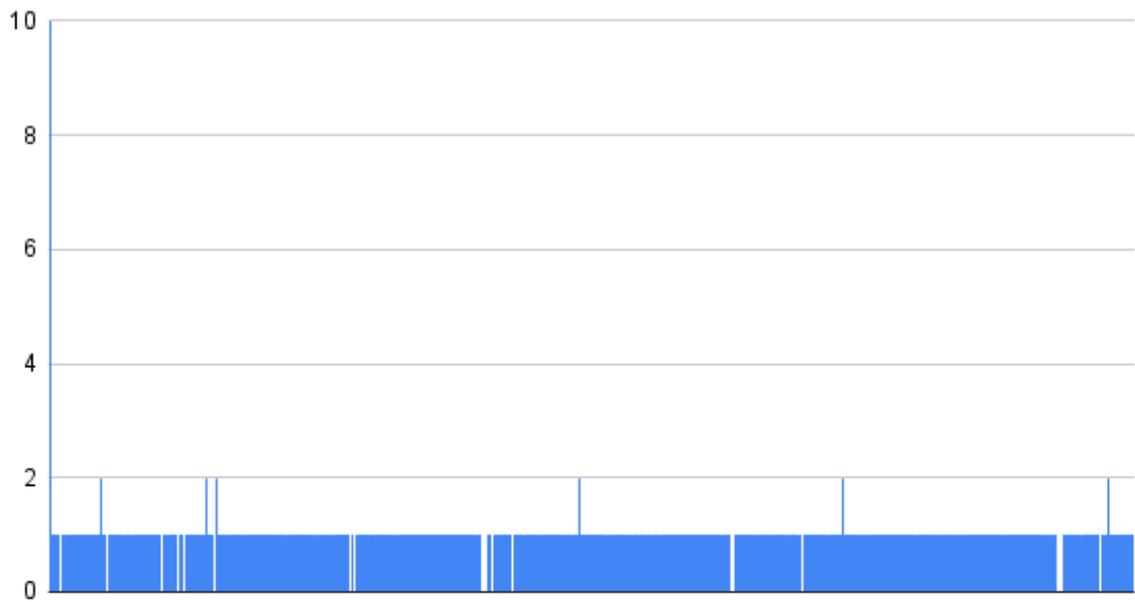


Figura 11 – Gráfico de tempo de execução de operações READ no SGBD PostgreSQL.

Relacional: Update

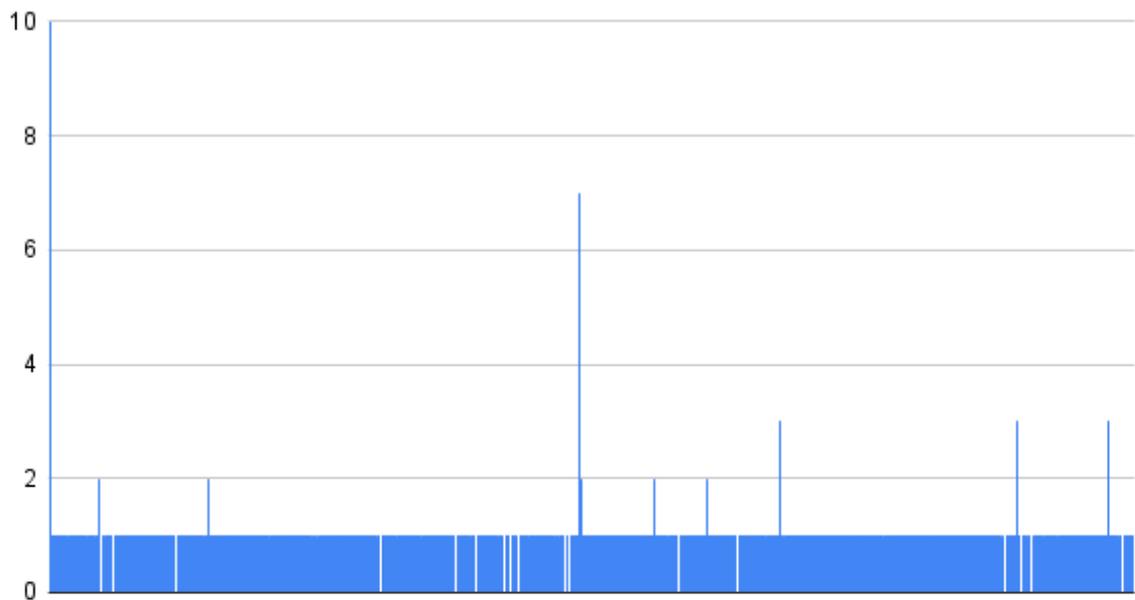


Figura 12 – Gráfico de tempo de execução de operações UPDATE no SGBD PostgreSQL.

Relacional: DELETE

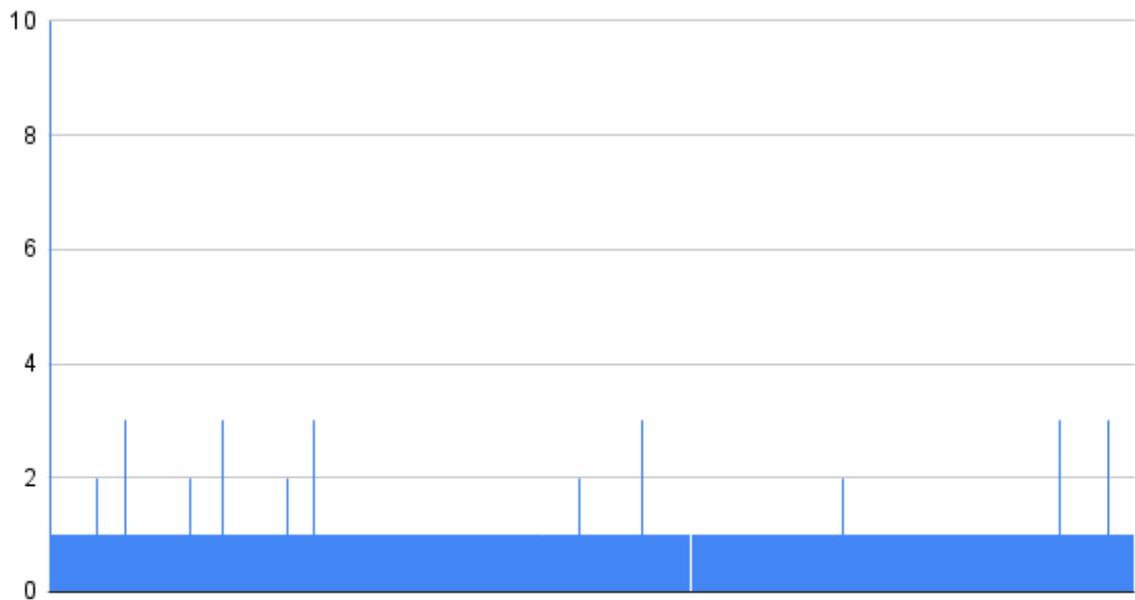


Figura 13 – Gráfico de tempo de execução de operações DELETE no SGBD PostgreSQL.

Relacional: Todos

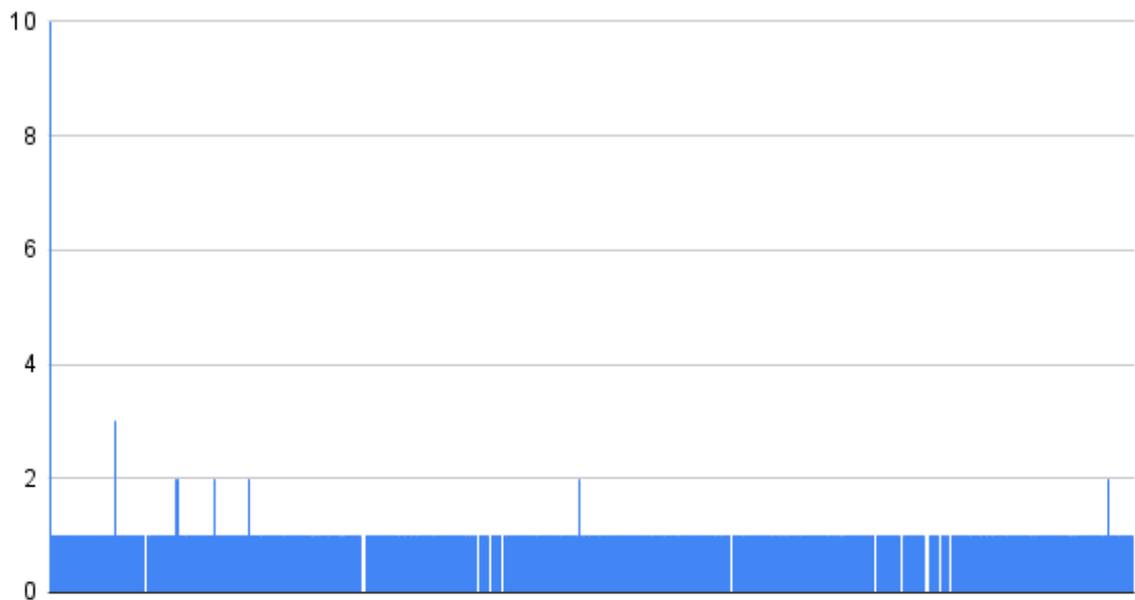


Figura 14 – Gráfico de tempo de execução de operações recuperando Todos os dados no SGBD PostgreSQL.

Relacional: 1:N

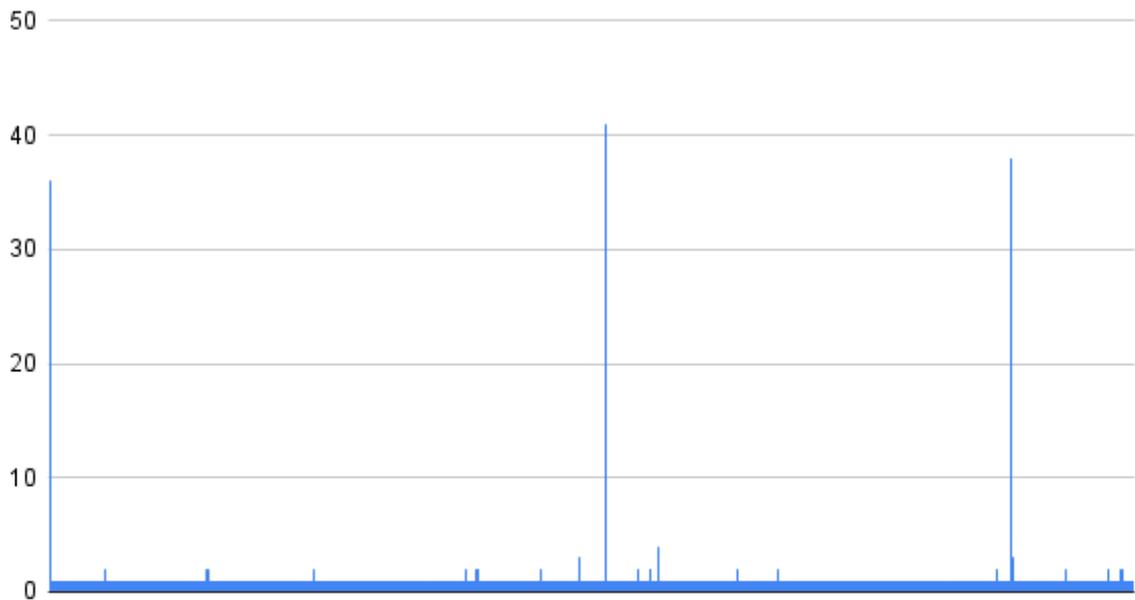


Figura 15 – Gráfico de tempo de execução de operações 1:N no SGBD PostgreSQL.

Relacional: N:N

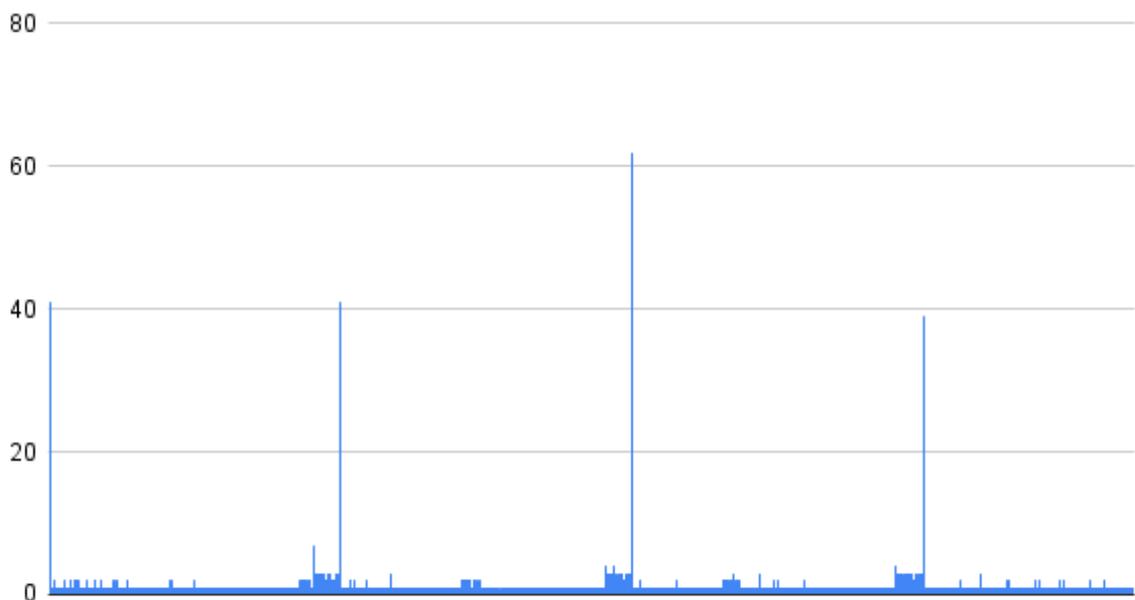


Figura 16 – Gráfico de tempo de execução de operações N:N no SGBD PostgreSQL.

É notável que os resultados obtidos demonstram que, dentro desse contexto, o banco de dados relacional teve um melhor desempenho, não só obtendo médias melhores,

mas sendo mais consistente com variação e desvio padrão menores dos que o do banco não relacional.

Comparando os gráficos das operações fica claro que ou o desempenho fica igual, em especial nas operações mais simples, ou o desempenho do banco de dados relacional é muito superior ao modelo não relacional. E mesmo nos casos em que o desempenho é semelhante, no modelo não relacional há muitos mais picos de tempo de execução, indicando uma inconsistência no modo com que o modelo lida com os dados.

O modelo não relacional em especial teve muitas dificuldades de lidar com a procura 1:N, e a procura de todos os dados da tabela, mantendo médias muito mais altas comparadas com modelo relacional. Outro ponto interessante é que nas procuras N:N ambos os modelos tiveram 3 picos muito semelhantes onde as procuras demoraram mais tempo, e nesse caso o maior pico foi do modelo relacional, porém a média do modelo relacional ainda foi menor.

5 Conclusão

O intuito desse trabalho era responder a pergunta que tive quando me propuseram o desafio de migrar os dados de uma base relacional para uma não relacional. A pergunta de que se, naquele caso específico e num ponto de vista de desempenho, o esforço de migrar os dados valeu a pena. Apesar das variadas opções serem bem vindas, a questão de "qual tecnologia usar" pode acabar virando um problema. Vários cenários bastante comuns para desenvolvedores que usam bancos de dados diariamente foram explorados nos testes, gerando dados que podem vir a ser úteis numa futura decisão de tecnologia para algum projeto.

Os resultados obtidos foram realmente uma surpresa. Na preparação e levantamento bibliográfico deste artigo muito se falou sobre a performance impressionante dos bancos de dados não relacionais. Os testes, porém, determinaram que o modelo relacional tem uma agilidade maior no conjunto de dados proposto, conseguindo ser constantemente mais rápido e mais constante que o modelo não relacional. Os motivos desse resultado podem ser variados, os dados fornecidos podem ainda ter um formato muito relacional, ou seja, um formato muito condizente a ser armazenado em tabelas. Isso faria com que o modelo relacional se adaptasse bem ao fazer o armazenamento e recuperação dessas informações. Outra questão a se considerar é que a quantidade de informações empregada nesta pesquisa não foi realmente massiva, uma quantidade massiva de dados provavelmente beneficiaria mais o modelo não relacional, que tem como foco a escalabilidade (TAMANE, 2016)

Dentro do escopo dessa pesquisa, considerando puramente a métrica de tempo, a resposta para a pergunta "relacional ou não relacional" claramente fica a favor do modelo relacional. Como apresentado nas figuras 3 e 4, o desempenho foi indiscutivelmente melhor, apresentando muitas médias com menos da metade do tempo de execução. Com esses resultados obtidos é seguro falar que, caso apresentado com uma situação semelhante, minha reação será manter os dados relacionais na base relaciona.

É claro que há outros fatores a se considerar, como a flexibilidade de modelos não relacionais, as promessas de escalabilidade e a facilidade de publicar um serviço como o MongoDB são questões que devem ser consideradas e analisadas. É importante considerar também que os dados e sua estrutura e tipo podem ter influenciado os resultados. Neste trabalho o escopo foi propositalmente pequeno, se limitando à minha pergunta inicial, mas é possível que com mais elementos que são notoriamente difíceis de lidar em bases relacionais (como imagens) o modelo não relacional teria sido superior. Cada projeto tem seu conjunto de necessidades que devem ser cuidadosamente analisadas antes de se tomar uma decisão que pode afetar tanto o seu andamento.

Em futuros trabalhos que abordem esse tema, os autores podem se beneficiar

em focar no uso de dados não relacionais ou semi relacionais, para aproveitar mais do poder das bases não relacionais. Outra mudança importante seria também o teste com quantidades maiores de informações para colocar a prova a escalabilidade dos modelos.

Referências

- ALVES, W. P. *Banco de dados*. [S.l.]: Saraiva Educação SA, 2014.
- BERG, K. L. et al. History of databases. *International Journal of Management & Information Systems (IJMIS)*, v. 17, n. 1, p. 29–36, 2013.
- CODD, E. F. A relational model of data for large shared data banks. *Communications of the ACM*, ACM New York, NY, USA, v. 13, n. 6, p. 377–387, 1970.
- COFFMAN, K.; ODLYZKO, A. The size and growth rate of the internet. Munksgaard International Publishers Ltd., Copenhagen, 1998.
- DATE, C. J. *Introdução a sistemas de bancos de dados*. [S.l.]: Elsevier Brasil, 2004.
- ELMASRI, R. et al. *Sistemas de banco de dados*. Pearson Addison Wesley São Paulo, 2005.
- GARCIA, V. S.; SOTTO, E. C. S. Comparativo entre os modelos de banco de dados relacional e não-relacional. *Revista Interface Tecnológica*, v. 16, n. 2, p. 12–24, 2019.
- GILBERT, S.; LYNCH, N. Perspectives on the cap theorem. *Computer*, IEEE, v. 45, n. 2, p. 30–36, 2012.
- GROFF, J. R.; WEINBERG, P. N.; OPPEL, A. J. *SQL: the complete reference*. [S.l.]: McGraw-Hill/Osborne, 2002. v. 2.
- GYORÖDI, C.; GYORÖDI, R.; SOTOC, R. A comparative study of relational and non-relational database models in a web-based application. *International Journal of Advanced Computer Science and Applications*, Science and Information (SAI) Organization Limited, v. 6, n. 11, p. 78–83, 2015.
- HOWS, D.; MEMBREY, P.; PLUGGE, E. *Introdução ao MongoDB*. [S.l.]: Novatec Editora, 2019.
- KLEINROCK, L. Distributed systems. *Communications of the ACM*, ACM New York, NY, USA, v. 28, n. 11, p. 1200–1213, 1985.
- KORTH HF, S. *Sistemas de Bancos de Dados*. [S.l.]: Makron Books, 1994.
- LEE, E. A. et al. Quantifying and generalizing the cap theorem. *arXiv preprint arXiv:2109.07771*, 2021.
- LÓSCIO, B. F.; OLIVEIRA, H. d.; PONTES, J. d. S. Nosql no desenvolvimento de aplicações web colaborativas. *VIII Simpósio Brasileiro de Sistemas Colaborativos*, sn, v. 10, n. 1, p. 11, 2011.
- MACÁRIO, C. G. d. N.; BALDO, S. M. O modelo relacional. *Instituto de Computação Unicamp. Campinas*, p. 1–15, 2005.
- MILANI, A. *PostgreSQL-Guia do Programador*. [S.l.]: Novatec Editora, 2008.
- PAREDAENS, J. et al. *The structure of the relational database model*. [S.l.]: Springer Science & Business Media, 2012. v. 17.

PART, A. Database management systems. 2006.

TAMANE, S. Non-relational databases in big data. In: *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*. [S.l.: s.n.], 2016. p. 1–4.

TATSIS, K. *A quantitative study on the popularity and performance of SQL and NoSQL DBMS*. 2022.