

# **Análise Comparativa de Desempenho entre Bases de Dados Relacional e Não Relacional Orientada a Documentos**

**Quézia Menezes Filadelfo<sup>1</sup>, Efânio Jeferson Amaral de Oliveira<sup>1</sup>,  
Clécio Silva Ferreira<sup>1</sup>, Pablo Freire Matos<sup>2</sup>**

<sup>1</sup>Discente de graduação em Sistemas de Informação. <sup>2</sup>Professor do Curso de Sistemas de Informação. IFBA *Campus* Vitória da Conquista  
e-mail: {queziafiladelfo, efaniojeferson, sfclecio, pablofmatos}@gmail.com

**RESUMO:** Este artigo objetiva investigar a capacidade de escalabilidade de um sistema gerenciador de banco de dados relacional em comparação ao banco de dados não relacional orientado a documentos. Desta forma, analisou-se o tempo de resposta de cinco consultas em uma base de dados real, com aproximadamente 5.300 registros e, posteriormente, com quatro diferentes tamanhos, gerados aleatoriamente (10, 100 e 500 mil, e 5 milhões de registros). Em seguida, comparou-se o resultado das consultas na base de dados do modelo relacional (MySQL) com os mesmos dados no modelo não relacional orientado a documentos (MongoDB) em duas modelagens distintas: referência para documentos e documentos embutidos. Nesse sentido, foram desenvolvidas três ferramentas que possibilitaram a realização de testes de *benchmark*, a fim de alcançar o objetivo do estudo. Os resultados apontaram uma superioridade na execução das consultas no modelo relacional em comparação ao modelo não relacional. No entanto, destaca-se que, à medida que o número de registros da base de dados aumenta, o MongoDB na estratégia de documentos embutidos mostra-se uma alternativa viável ao modelo relacional, sendo mais rápido em um dos cenários testados.

**PALAVRAS-CHAVE:** armazenamento, desempenho, escalabilidade

## **Comparative Analysis of Performance between Relational Database and Document-Oriented Non-Relational Database**

**ABSTRACT:** This paper aimed to investigate the scalability of a relational database management system compared to a document-oriented non-relational database. In this way, the response time of five queries were analyzed in a real database, with approximately 5,300 records and, later, with four different sizes, generated randomly (10, 100 and 500 thousand, and 5 million records). Then, the result of the relational database in MySQL was compared with the same document-oriented non-relational database in MongoDB in two different models: reference for documents and embedded documents. In this sense, three tools were developed that made it possible to carry out benchmark tests in order to achieve the objective of the study. The results showed a superiority in the execution of queries in the relational database compared to the non-relational database. However, it highlights that, as the number of records in the database increases, MongoDB in the embedded documents strategy proves to be a viable alternative to the relational model, being faster in one of the tested scenarios.

**KEYWORDS:** storage, performance, scalability

## INTRODUÇÃO

Desde a sua criação até os dias atuais, os bancos de dados relacionais têm sido amplamente utilizados pela grande maioria das aplicações e sistemas, atuando como sucessores aos modelos hierárquicos e de redes, por conseguir resolver diversos problemas como a falta de consistência e de indisponibilidade dos dados. A justificativa para o grande uso do modelo relacional se dá pela garantia de integridade, disponibilidade e consistência provindas das propriedades de transações ACID (Atomicidade, Consistência, Isolamento e Disponibilidade), além da facilidade de integração com as mais diversas linguagens de programação (GRAY, 1981). Entretanto, devido ao crescente volume de dados gerado pelas aplicações, bem como dos diferentes tipos de dados e dos diversos sistemas distribuídos em rede, em distintos casos, estes aspectos podem se constituir como limitadores. Exame (2021) destaca que, só no último ano, foram gerados cerca de 40 trilhões de gigabytes de dados. Diante do exposto, pode-se inferir que o uso do modelo relacional em ambientes tão dinâmicos, pode não se mostrar tão eficaz.

Diante disso, os bancos de dados não relacionais tornam-se fundamentais, apresentando-se como uma alternativa ao modelo relacional, ao permitir o processamento dos dados de forma rápida (ANICETO; XAVIER, 2014). Estes também têm como foco principal o desempenho, no qual é possível modelar dados sem seguir a estrutura rígida do modelo relacional, além de lidar com uma arquitetura distribuída e tolerante a falhas. Presume-se, então, que os bancos de dados não relacionais, por não possuírem uma estrutura tão rígida quando comparados aos bancos de dados relacionais, podem prover uma maior eficiência a um sistema, tendo em vista que proporcionam diversas formas de armazenamento de dados, tais como documentos, grafos e colunas, além de não seguir as propriedades ACID e a linguagem SQL (Linguagem de Consulta Estruturada), em troca de velocidade, disponibilidade e capacidade inerente de escalabilidade dos dados (SADALAGE; FOWLER, 2013).

Apesar dos bancos de dados relacionais serem os mais utilizados, desde o seu desenvolvimento, e tidos como padrão quando se fala em armazenamento de dados, em diversos pontos o modelo relacional se mostra ineficiente, por exemplo, na disponibilidade de dados, no desempenho, na capacidade de lidar com alto volume de dados e na capacidade de escalabilidade (DIANA; GEROSA, 2010). Atualmente, conforme Sadalage e Fowler (2013), o aumento dos dados gerados pelas aplicações exige muito mais recursos computacionais, para que seja possível atender a demanda.

Diana e Gerosa (2010) destacam que a principal desvantagem de aplicar a estratégia de bancos de dados relacionais é a perda de parte da sua capacidade de lidar com restrições de dados, além de não serem mais capazes de realizar JOINS (junção de duas ou mais tabelas) transparentemente. Sadalage e Fowler (2013) ainda destacam outros dois pontos que agem como limitadores ao modelo relacional: o mapeamento de dados, pois demanda muito tempo de desenvolvimento das aplicações; e o fato de os bancos de dados relacionais serem projetados para rodar em uma única máquina, existindo, assim, uma limitação ao se processar grandes quantidades de dados. Baseado nas limitações do modelo relacional apresentadas, este artigo propõe um estudo que objetiva comparar o desempenho entre o modelo relacional e o modelo não relacional orientado a documentos.

Em face do exposto, o presente trabalho apresenta uma análise comparativa do desempenho de consultas de uma base de dados relacional confrontadas com as mesmas consultas realizadas em uma base de dados não relacional. Para tanto, foram desenvolvidas três ferramentas que possibilitaram a realização de testes de benchmark: (i) a primeira teve o objetivo de povoar a base de dados relacional com dados aleatórios; (ii) a segunda teve o objetivo de migrar os dados da base de dados do modelo relacional para o modelo não relacional orientado a documentos; (iii) e a terceira foi responsável em

contabilizar o tempo de resposta das consultas. Posteriormente, são apresentados os resultados obtidos em cada cenário, a fim de evidenciar em quais cenários uma base de dados é, de fato, superior a outra no ambiente de teste proposto.

Nesse sentido, o restante deste artigo está estruturado da seguinte forma: Na Seção 2, são discutidos os materiais e métodos utilizados neste artigo, na Seção 3, são apresentados os resultados e as discussões da análise comparativa, e na Seção 4, são apresentadas as conclusões e os trabalhos futuros.

## MATERIAL E MÉTODOS

A primeira etapa do trabalho consistiu em realizar o levantamento teórico acerca dos bancos de dados relacionais e não relacionais e suas principais características. Para tornar possível a realização da pesquisa proposta, foi utilizado como estudo de caso a base de dados do Sistema de Certificados (RAMOS *et al.*, 2018). O sistema é utilizado para a emissão e a validação de certificados em eventos acadêmicos realizados no Instituto Federal de Educação, Ciência e Tecnologia da Bahia, *campus* Vitória da Conquista, desde o ano de 2017. O sistema foi desenvolvido utilizando a linguagem de programação PHP com o *framework* Zend e com acesso aos dados por meio do sistema gerenciador de bando de dados relacional MySQL.

Na Figura 1 são mostradas as etapas necessárias para realizar a análise comparativa de desempenho entre as bases de dados relacional e não relacional orientada a documentos. Na Etapa 1 foi criada uma ferramenta de povoamento de dados aleatórios a fim de povoar o banco de dados relacional. Como a base de dados do Sistema de Certificados contém uma quantidade pequena de dados, faz-se necessário a criação de um mecanismo de povoamento aleatório de dados para comportar uma grande quantidade de dados. A ferramenta foi desenvolvida na linguagem de programação Python, utilizando a biblioteca Faker.

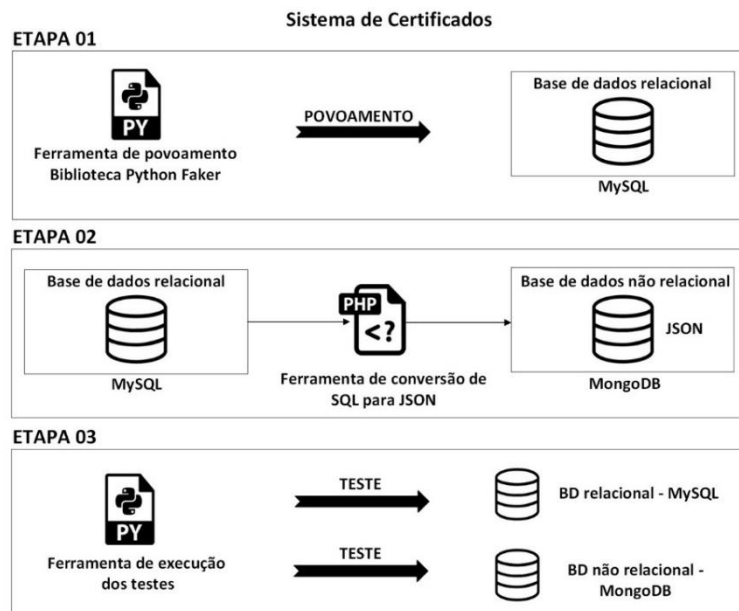


Figura 1. Etapas realizadas. IFBA, 2022.

Na Etapa 2 consistiu em realizar a migração da base de dados do modelo relacional em SQL para o modelo não relacional orientado a documentos em JSON. A forma dos bancos de dados orientados a documentos armazenarem seus dados se assemelha aos de chave valor, onde se tem um conjunto de documentos, cada um deles possuindo campos e os valores destes campos. Foi desenvolvido uma ferramenta própria utilizando

a linguagem de programação PHP para possibilitar essa migração, que conta com uma série de scripts, onde cada script atua convertendo as tabelas da base de dados do formato SQL para JSON.

Na Etapa 3 procedeu-se a definição de um plano de teste, com o auxílio de uma ferramenta desenvolvida em Python que é responsável em contabilizar o tempo de resposta das consultas. Foram utilizadas cinco consultas, a saber: C1) Qual o total de atividade por evento?, C2) Qual o total de participante por evento?, C3) Quais as atividades e tipos de atividade de um evento?, C4) Qual o total de certificado por evento? e C5) Quais os eventos de um participante?. O objetivo do teste de carga foi validar o comportamento sob as mesmas condições normais de carga.

## RESULTADOS E DISCUSSÃO

Como resultados foram criadas três ferramentas, a primeira responsável pelo povoamento aleatório na base de dados relacional, disponível em <https://github.com/certificadosifba/Ferramenta-Povoamento-Aleatorio>; a segunda é responsável por fazer a migração de uma base dados relacional para uma não relacional orientada a documentos disponível em <https://github.com/certificadosifba/Ferramenta-Conversao-SQL-JSON>; e a terceira é encarregada da execução de testes que calcula o tempo de resposta de consultas em uma base de dados relacional e em uma não relacional orientada a documentos disponível em <https://github.com/certificadosifba/Ferramenta-Execucao-de-Testes/>.

Os testes foram executados em uma única máquina, rodando o sistema operacional Windows 11 com arquitetura de 64 bits, equipada com um processador Intel Core I5-11400H com clock de até 4.50 Ghz, 8GB de memória RAM DDR4 com frequência de 3200MHz, e uma placa de vídeo dedicada NVIDIA RTX 3050 com 4GB GDDR6. Em relação aos sistemas gerenciadores de banco de dados utilizados, para o modelo relacional foi utilizado o MySQL Community Server na versão 8.0.28, e para o modelo não relacional, o MongoDB Community Server na versão 5.0.6.

Os testes foram realizados utilizando a mesma consulta em ambas bases de dados. O tempo de execução foi computado na mesma consulta cinco vezes, descartando o pior e o melhor resultado, a fim de evitar quaisquer variações provenientes de picos ou lentidão durante a execução das consultas. Posto isto, o tempo de execução foi calculado utilizando a média aritmética. O primeiro cenário de teste (Tabela 1) ao utilizar dados contidos na base de dados original, com cerca de 5.300 registros, apresenta melhores resultados nas consultas realizadas em MySQL do que as modelagens utilizadas no MongoDB.

**Tabela 1.** Testes com Dados Originais (5.300 registros). IFBA, 2022.

| Consulta | MySQL  | Mongo DBRef | Mongo Embedded |
|----------|--------|-------------|----------------|
| 1        | 0,0026 | 0,0340      | 0,0282         |
| 2        | 0,0105 | 0,1229      | 0,0234         |
| 3        | 0,0051 | 0,0673      | 0,0245         |
| 4        | 0,0048 | 0,1210      | 0,0264         |
| 5        | 0,0022 | 0,0142      | 0,0103         |

No segundo cenário apresentado na Tabela 2, com consultas de 10 e 100 mil registros, o MySQL ainda manteve melhores resultados, com exceção da Consulta 2, no qual com a modelagem de documentos embutidos (*Embedded Documents*) o MongoDB apresentou-se cinco vezes mais rápido que o modelo relacional quando utilizados 100 mil registros.

**Tabela 2.** Testes com 10 e 100 mil registros. IFBA, 2022.

| Consulta | 10.000 |             |                | 100.000 |             |                |
|----------|--------|-------------|----------------|---------|-------------|----------------|
|          | MySQL  | Mongo DBRef | Mongo Embedded | MySQL   | Mongo DBRef | Mongo Embedded |
| 1        | 0,0029 | 0,0279      | 0,1630         | 0,0030  | 0,0258      | 1,2890         |
| 2        | 0,1202 | 0,8387      | 0,1106         | 8,7962  | 8,2877      | 1,6289         |
| 3        | 0,0031 | 0,0809      | 0,1970         | 0,0039  | 0,0769      | 1,5416         |
| 4        | 0,0142 | 0,8338      | 0,1477         | 0,7227  | 5,8717      | 1,1630         |
| 5        | 0,0026 | 0,0277      | 0,0428         | 0,0027  | 0,1061      | 0,0922         |

No terceiro cenário (Tabela 3), foram utilizados 500 mil e 5 milhões de registros, o MySQL se manteve superior ao modelo não relacional, perdendo novamente na Consulta 2. O MongoDB com a modelagem *Embedded Documents* foi duas vezes mais rápido com 500 mil registros e três vezes mais rápido com 5 milhões de registros.

**Tabela 3.** Testes com 500 mil e 5 milhões de registros. IFBA, 2022.

| Consulta | 500.000 |             |                | 5.000.000 |             |                |
|----------|---------|-------------|----------------|-----------|-------------|----------------|
|          | MySQL   | Mongo DBRef | Mongo Embedded | MySQL     | Mongo DBRef | Mongo Embedded |
| 1        | 0,0035  | 0,0255      | 4,0485         | 0,0024    | 0,0266      | 15,8320        |
| 2        | 13,392  | 28,7846     | 4,6558         | 50,596    | 63,9515     | 17,6758        |
| 3        | 0,0037  | 0,0899      | 4,9523         | 0,0024    | 0,0853      | 14,0339        |
| 4        | 0,7081  | 35,662      | 3,6187         | 8,2209    | 56,2658     | 11,6542        |
| 5        | 0,0031  | 0,3944      | 0,3221         | 0,0019    | 0,8512      | 1,9476         |

Através desses resultados é possível constatar que houve superioridade em termos de velocidade de execução das consultas do modelo relacional em praticamente todos os cenários, tendo apenas como ressalva a Consulta 2, que através da estratégia *Embedded Documents* foi mais rápida no modelo não relacional. Isso demonstra que apesar de ser mais lento, utilizando a modelagem *Embedded Documents*, o MongoDB se demonstra mais estável conforme o número de registros de dados aumenta. Já o relacional apresentou perda de desempenho em relação ao tempo de resposta conforme o aumento de dados, particularmente na Consulta 2, o que requer uma investigação futura.

Ao observar os resultados, também ficou visível que o modelo utilizando DBRef (Referência para Documentos) foi o mais lento dentre todos, constatando assim que o particionamento de uma base de dados em várias coleções acarreta em uma lentidão maior do que o esperado na recuperação de dados. Contudo, essa modelagem pode ser útil se houver um maior agrupamento de dados que são frequentemente utilizados, minimizando assim, consultas que utilizem referências de outras coleções.

É importante ressaltar o espaço ocupado em disco em cada um dos modelos. No cenário utilizando 5 milhões de registros, a base de dados relacional ocupou 1,6 gigabytes de espaço, enquanto o MongoDB com a modelagem DBRef ocupou cerca de 500 megabytes, e com a modelagem *Embedded Documents* foram utilizados

aproximadamente 830 megabytes. Desta forma, em um cenário onde há restrição de espaço para armazenamento de dados, o modelo não relacional se faz uma alternativa.

## CONCLUSÕES

Através dos resultados obtidos conclui-se que o estudo realizado nesse artigo pode ser útil para determinar qual base de dados utilizar na construção de determinados sistemas, levando em conta a demanda do sistema e suas restrições. Como contribuições tem-se o desenvolvimento de uma ferramenta de migração SQL para JSON, que apesar de ter sido feita sob demanda para o sistema utilizado, pode ser reaproveitado para outras bases de dados. Também foram geradas bases de dados durante este estudo que podem ser utilizadas em pesquisas futuras e o desenvolvimento de uma ferramenta de execução de testes, cujos resultados obtidos podem servir como parâmetro para ajudar a determinar qual sistema gerenciador de dados utilizar em determinados sistemas. As bases de dados utilizadas encontram-se disponíveis em [https://drive.google.com/drive/folders/1qu-JsCLYk7s5\\_6jx77zhHO-yIjsix2u3-?usp=sharing](https://drive.google.com/drive/folders/1qu-JsCLYk7s5_6jx77zhHO-yIjsix2u3-?usp=sharing).

Como trabalhos futuros propõem-se as seguintes melhorias: (i) usar indexação na base de dados relacional e não relacional; (ii) utilizar uma maior quantidade de dados nos testes, por exemplo, 10, 50 e 100 milhões de dados; (iii) executar testes com outras operações DML nas bases de dados; (iv) desenvolver uma aplicação que consiga migrar de uma forma mais simples uma determinada base de dados relacional para uma não relacional, seguindo um modelo de regra estabelecido em um arquivo JSON e (v) executar testes de estresse até o MySQL se tornar mais lento e comparar com o desempenho do MongoDB.

## REFERÊNCIAS

ANICETO, R. C.; XAVIER, R. F. **Um Estudo sobre a Utilização do Banco de Dados NoSQL Cassandra em Dados Biológicos**. 2014. 50 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Instituto de Ciências Exatas, Universidade de Brasília, Brasília, 2014.

DIANA, M.; GEROSA, M. A. NoSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0. In: WORKSHOP DE TESTES E DISSERTAÇÕES EM BD (WTDB), XI, 2010, Belo Horizonte. **Anais...** Belo Horizonte: SBC, 2010.

EXAME. **Temos mais dados do que nunca**. Como usá-los a nosso favor? 2021. Disponível em: <https://exame.com/carreira/dados-uso-favor/>. Acesso em: 19 ago. 2022.

GRAY, Jim. The Transaction Concept: Virtues and Limitations. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 7 th, 1981, Cannes, France. **Proceedings...** Cannes: VLDB, 1981, p. 144-154.

RAMOS, L. L.; SILVA, J. P.; SOBREIRA, A. D.; MATOS, P. F. Sistema Web e Open Source de Gerenciamento de Emissão e Validação de Certificado nos Institutos Federais. In: CONGRESSO NORTE NORDESTE DE PESQUISA E INOVAÇÃO (CONNEPI), XII, 2018, Recife, PE. **Anais...** Recife, 2018. p. 1-10.

SADALAGE, P.; FOWLER, M. **NoSQL Essencial: Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**. São Paulo: Novatec, 2013. 216 p.