VI Simpósio de Iniciação Científica 2023
Ciência como ferramenta de transformação da sociedade

Armazenamento do Modelo Orientado a Objetos em Estrutura Orientada a Coluna

Ismael Souza Alves[1] (IC), Enzo Seraphim (PQ)[1]
[1] *Universidade Federal de Itajubá - UNIFEI.*

Palavras-chave: orientado a coluna. análise de desempenho. banco de dados.

Introdução

A tecnologia digital está presente em muitas tarefas do cotidiano na produção de arquivos como, por exemplo, de documentos, de fotos, ou de vídeos. De acordo com Dell (2023) e seus dados de pesquisa, o volume de dados gerados anualmente crescerá a uma taxa de 21,2% e de 84 mil exabytes em 2021, atingindo um total de 221 mil exabytes até 2026. Para que esses arquivos estejam disponíveis a todo momento é necessário o armazenamento em disco. Existem diversas formas de armazenamento de dados em disco, sendo comum o uso de Sistema Gerenciador de Banco de Dados (SGBD).

Dentre os diversos tipos de SGBDs, de acordo com a pesquisa da SolidIT (2023), em agosto de 2023, os bancos relacionais representam 71,9% de todos os SGBDs utilizados, pois garantem atomicidade, consistência, isolamento, durabilidade (ELMASRI RAMEZ; NAVATHE, 2019) . A atomicidade em uma transação que envolve duas ou mais partes garante que será executada totalmente ou não será executada. Consistência em uma transação cria um novo estado válido dos dados ou retorna todos os dados ao seu estado antes que a transação seja iniciada. Isolamento em uma transação em andamento não será interferida por nenhuma outra transação concorrente. A durabilidade garante que os dados validados pelo sistema estão disponíveis em seu estado correto mesmo em caso de falha. Para acesso às informações, os SGBDs relacionais utilizam uma linguagem padronizada chamada “Structured Query Language” (SQL).

Na atualidade, novas formas baseadas em tecnologias de SGBDs NoSQL vêm se destacando (FOWLER, 2015), pois oferecem grande desempenho e escalabilidade para em grande quantidade de dados. Existem 3 tipos de SGBDs NoSQL (STONEBRAKER; ÇETINTEMEL; ZDONIK, 2010): orientado a coluna, orientado a grafos, ou orientado a documentos. A utilização de bancos de dados orientados a colunas vem crescendo já que são otimizados para a leitura de dados (ABADI; BONCZ; HARIZOPOULOS, 2008). O objetivo deste projeto é o armazenamento de um modelo orientado a objetos em uma estrutura orientada a colunas.

Metodologia

Foram definidas três etapas como metodologia, para o desenvolvimento deste trabalho: 1ª etapa: criação do modelo do experimento para base relacional e orientada à coluna; 2ª etapa: inserção de dados na base relacional e orientada à coluna em três cenários; 3ª etapa: geração de consultas idênticas para ambas bases de dados para os três cenários.

Para realização da primeira etapa, criação do modelo de aplicação, houve a necessidade de estabelecer compatibilidade no armazenamento de dados para o modelo relacional (SILBERSCHATZ A.; KORTH, 2011) e o modelo orientado a colunas (MCCREARY D.; KELLY, 2013) (HARIZOPOULOS et al., 2009), ou seja, ambas bases de dados deveriam ter a mesma informação para que as consultas não sofressem alterações para efeito de comparação.

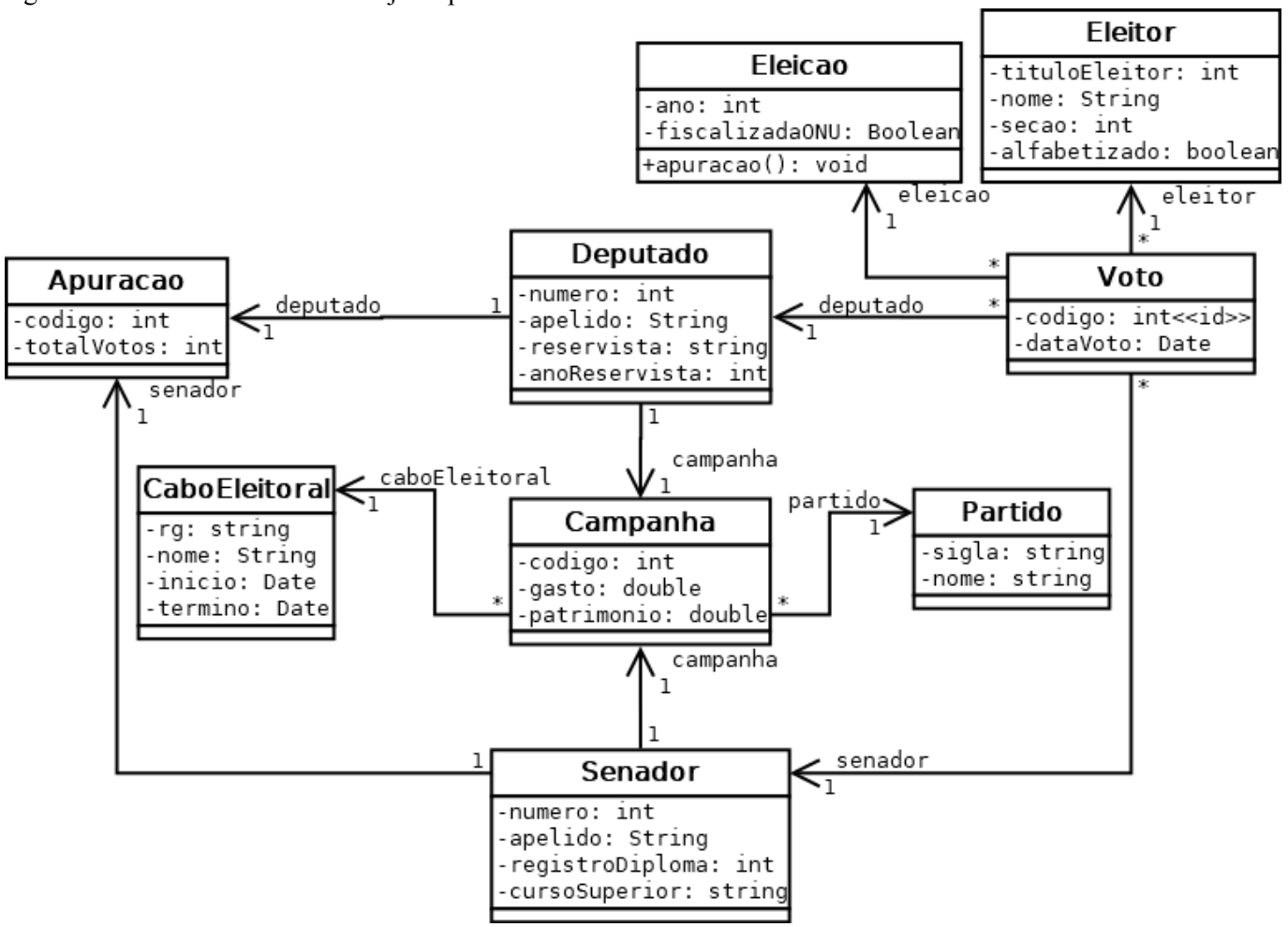
Para o desenvolvimento do projeto foi escolhido um modelo que simula o comportamento de eleições onde eleitores votam para deputados e senadores que terão suas respectivas apurações de votos. Este modelo de aplicação inclui informações relacionadas a eleitores, candidatos, eleições, apuração de votos, campanhas eleitorais, partidos políticos e cabo eleitoral.

Foi elaborado um diagrama UML, conforme ilustrado na Figura 1, que descreve um modelo de aplicação orientada a objetos que será armazenada em base de dados relacional e orientada à coluna.

Este modelo de aplicação foi criado para simular um ambiente de eleições e votações, incluindo informações relacionadas a eleitores, candidatos, eleições, apuração de votos, campanhas eleitorais, partidos políticos e cabo eleitoral. O modelo de dados proposto permitiu realizar consultas que foram posteriormente usadas nos experimentos para avaliar o desempenho dos sistemas de gerenciamento de bancos de dados em cenários de diferentes tamanhos de eleições. Essa abordagem garantiu que as consultas fossem consistentes e comparáveis entre o modelo relacional e o modelo orientado à colunas.

Para a segunda etapa, foi realizada a implementação de uma aplicação em Java (DEITEL; DEITEL, 2005) do modelo, a fim de garantir a consistência dos dados e medir a escalabilidade com diversos volumes de dados.

Figura 1 – Modelo Orientado a Objetos para base de dados relacional e orientada à coluna.



A base de dados utilizada nesta pesquisa consiste em preencher com informações as variáveis não numéricas do modelo proposto, bem como facilitar na interpretação do banco e aproximar os resultados de um caso real.

Inicialmente a inserção de dados na base relacional totalizou 4.675 objetos distribuídos entre as seguintes classes: 1.131 Eleitores; 1.131 Votos; 81 Senadores; 515 Deputados; 596 Campanhas; 596 Cabos Eleitorais; 596 Apurações; 28 Partidos Eleitorais e 1 Eleição.

Para atributos do tipo cadeia de caracteres, utilizou-se os mesmos valores que foram obtidos de dados públicos coletados da internet, sendo que, para: Eleitores utilizou-se nomes próprios únicos da internet; e Senadores, Deputados e Partidos da eleição de 2006.

Para evitar influência de sequências numéricas todos atributos numéricos foram gerados aleatoriamente. A geração de dados considerou que todos eleitores votam na eleição, sendo que o seu voto para senador e para deputados foi escolhido aleatoriamente. Todo senador e deputado teve uma Campanha com Apuração na Eleição, mas seu único o Cabo Eleitoral foi escolhido de forma aleatória.

Para essa base relacional foi estabelecido 3 cenários de inserção de dados:

- Primeiro cenário contendo 10 eleições totalizando 46.498 registros;
- Segundo cenário contendo 100 eleições totalizando 464.728 registros;
- Terceiro cenário contendo 1.000 eleições totalizando 4.647.028 registros;

Diferentemente, a inserção de dados na base orientada a colunas totalizou 1.131 registros, pois nesta abordagem os dados são armazenados em apenas uma tabela.

Para essa base orientada a colunas foi estabelecido 3 cenários de inserção de dados:

- Primeiro cenário contendo 10 eleições totalizando 11.310 registros;
- Segundo cenário contendo 100 eleições totalizando 113.100 registros;
- Terceiro cenário contendo 1.000 eleições totalizando 1.131.000 registros;

Para garantir menor interferência do meio utilizado, o modelo relacional e o modelo orientado a coluna foram armazenados no banco MariaDB (DYER, 2015).
Para a terceira etapa, geração de consultas idênticas, foram geradas todas combinações de 2 atributos de classes diferentes pertencentes ao modelo. Considerando que o modelo tem 36 atributos, foram gerados 630 consultas com pares de atributos diferentes que foram executadas na base de dados relacional e na base de dados orientada a coluna.
Para avaliar o desempenho optou-se em medir o tempo de execução entre o início da execução e seu término em cada consulta no banco de dados.
Com o objetivo de minimizar a influência de oscilações de hardware nos resultados, as consultas foram repetidas 6 vezes e contabilizado a média de tempo de execução.

Resultados e discussão

Após medir o tempo de execução das consultas no modelo orientado a coluna e no modelo relacional, foram gerados três gráficos com os cenários de 10 eleições (figura 2), 100 eleições (figura 3) e 1.000 eleições (figura 4).

Figura 2 – Desempenho de consulta em 10 eleições

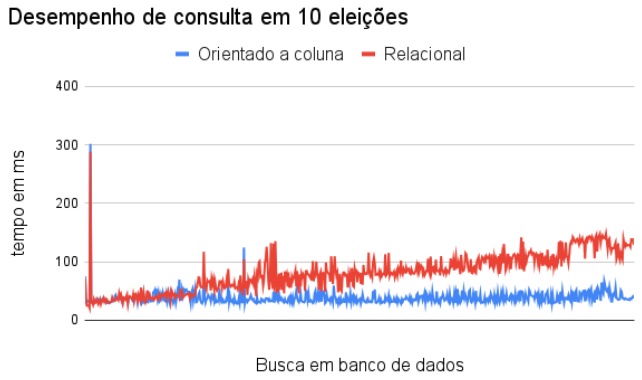


Figura 3 – Desempenho de consulta em 100 eleições

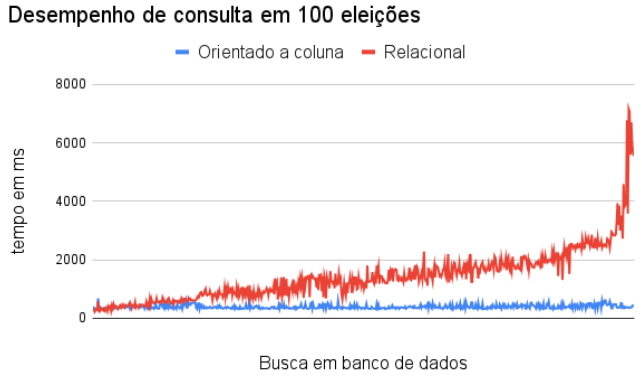


Figura 4 – Desempenho de consulta em 1000 eleições

Os resultados nos gráficos foram ordenados de forma crescente no eixo das consultas tendo como referência os valores do gráfico 4. Isso permitiu uma visualização de tendências de desempenho à medida que o número de eleições aumentava.
Os resultados da pesquisa revelaram a notável diferença de desempenho entre os modelos de banco de dados colunar e relacional em relação ao tempo de processamento das consultas. Ao comparar os tempos médios de busca em três cenários distintos, ficou evidente que o modelo colunar superou consistentemente o modelo relacional em tempo.
Para o gráfico da figura 1 com 10 eleições, o modelo colunar mostrou-se 2,11 vezes mais rápido do que o modelo relacional. O tempo médio de 630 consultas no modelo orientado a coluna foi 38,44 ms, enquanto o modelo relacional apresentou um tempo médio de 81,19ms.
O gráfico da figura 2 com 100 eleições a diferença de desempenho tornou-se ainda mais evidente. O modelo colunar superou o modelo relacional em 3,69 vezes, com um tempo médio de 391,33ms por busca, em comparação com os 1.421,05ms do modelo relacional. Essa diferença significativa já demonstra a vantagem do modelo orientado a coluna em ambientes que exigem respostas rápidas a consultas.
Finalmente, o gráfico da figura 3 com 1000 eleições o modelo colunar continuou a demonstrar sua superioridade, sendo 5,59 vezes mais rápido do que o modelo relacional. O tempo médio de busca no modelo colunar foi de apenas 3.672,57 ms, enquanto o modelo relacional exigiu 20.432,84 ms em média por busca. Isso ressalta a escalabilidade do modelo colunar quando se lida com um volume maior de consultas.

Conclusões

A análise comparativa dos gráficos demonstra claramente que o modelo colunar supera o modelo relacional em termos de desempenho e escalabilidade quando se trata de realizar consultas de banco de dados. Essa conclusão é de grande relevância em ambientes corporativos, onde a eficiência do processamento de dados desempenha um papel crítico em várias facetas das operações.

Em primeiro lugar, a diferença no tempo de execução das consultas entre os dois modelos de bancos de dados pode influenciar diretamente a tomada de decisões de mercado. Em um mundo empresarial cada vez mais orientado por dados, a rapidez com que informações críticas podem ser acessadas e analisadas é crucial. Um modelo de banco de dados que permite consultas mais rápidas e eficientes pode fornecer uma vantagem competitiva significativa, permitindo que as empresas reajam mais rapidamente às mudanças no ambiente de mercado.

Além disso, a experiência do usuário é afetada diretamente pelo desempenho do sistema. Se um aplicativo ou site empresarial depende de consultas frequentes ao banco de dados, um modelo colunar eficiente pode garantir uma experiência mais fluida e responsiva para os usuários. Isso é especialmente importante em setores onde a interação do cliente desempenha um papel fundamental, como comércio eletrônico, serviços financeiros online e muitos outros.

Portanto, a escolha do sistema de gerenciamento de banco de dados (SGBD) deve ser uma decisão estratégica cuidadosa em ambientes corporativos. O desempenho e a escalabilidade oferecidos pelo modelo colunar podem se traduzir em vantagens competitivas, eficiência operacional e uma experiência do usuário melhorada. No entanto, é importante lembrar que a escolha do SGBD também deve levar em consideração outros fatores, como a estrutura e a natureza dos dados, requisitos de segurança e conformidade, entre outros, para garantir uma solução completa e eficaz.

Agradecimentos

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo apoio financeiro que tornou este trabalho possível.

Ao meu orientador, professor Enzo Seraphim, cuja orientação foi fundamental para a concepção e desenvolvimento deste trabalho e seus conselhos inestimáveis para a qualidade deste artigo.

A minha família e amigos pelo apoio emocional e encorajamento ao longo deste processo, que foi fundamental para superar desafios e obstáculos durante esta jornada.

A Universidade Federal de Itajubá e seus colaboradores, cujo empenho contribuiu para o enriquecimento do conhecimento necessário para a elaboração desta pesquisa.

Ao Grupo de Pesquisas em Engenharia de Sistemas e de Computação (GPESC) da UNIFEI pelo apoio na realização desta pesquisa.

Referências

ABADI, D. J.; BONCZ, P. A.; HARIZOPOULOS, S. Column-stores vs. row-stores: How different are they really? In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. [S.l.: s.n.], 2008. p. 967–980.

DEITEL, H.; DEITEL, P. Java: como programar. Pearson Prentice Hall, 2005. ISBN 9788576050193.

DELL. High Data Growth and Modern Applications Drive New Storage Requirements in Digitally Transformed Enterprises. 2022. IDC Research. Disponível em: <https://www.delltechnologies.com/asset/en-my/products/storage/industry-market/h19267-wp-idc-storage-reqs-digital-enterprise.pdf>. Acesso em: 25 ago 2023.

DYER, R. J. Learning MySQL and MariaDB:. [S.l.]: O'Reilly Media, Inc., 2015.

ELMASRI RAMEZ; NAVATHE, S. B. Sistemas de banco de dados. 7. ed. [S.l.]: editora Pearson Education, 2019. ISBN 9788576050193.

FOWLER, A. NoSQL For Dummies. Hoboken: John Wiley Sons, 2015. ISBN 978-1-118-90562-3.

HARIZOPOULOS, S. et al. Column-oriented database systems. Proceedings of the VLDB Endowment, v. 2, n. 2, p. 1664–1665, 2009.

HARRINGTON, J. L. Relational Database Design and Implementation. Boston: Elsevier Inc., 2009. ISBN 978-0-12-374730-3.

MCCREARY D.; KELLY, A. Getting Started with NoSQL. Birmingham: Packt Publishing, 2013. ISBN 978-1-84969-4-988.

SOLIDIT, C. e desenvolvimento de software. Popularidade do DBMS dividida por modelo de banco de dados. 2023. Base de conhecimento de sistemas de gerenciamento de banco de dados relacionais e NoSQL. Disponível em: <https://db-engines.com/en/ranking>. Acesso em: 25 ago 2023.

SILBERSCHATZ A.; KORTH, H. F. S. S. Database System Concepts. 6. ed. Nova Iorque: The McGraw-Hill Companies, 2011.

STONEBRAKER, M.; ÇETINTEMEL, U.; ZDONIK, S. Nosql data stores. In: Proceedings of the 21st International Conference on Database and Expert Systems Applications. [S.l.: s.n.], 2010. p. 363–366.