



Eficiência de Consultas no SQL SERVER: Um estudo de caso por meio de análise de indexações

Gregory M. FAGUNDES¹; Hiran N. M. FERREIRA²

RESUMO

Este texto apresenta o resultado de uma análise de otimização de consultas SQL em um ambiente real, com foco na técnica de indexação para aprimorar o desempenho dessas consultas. O estudo foi conduzido por meio de experimentos com consultas SQL que não apenas proporcionam respostas rápidas, mas também são altamente eficientes para o usuário, priorizando consultas que retornam um grande volume de dados em um intervalo de tempo específico e consultas complexas que envolvem relações entre várias tabelas. Os resultados obtidos indicam que a otimização de consultas SQL através da técnica de indexação pode melhorar significativamente a eficiência e a velocidade de recuperação de dados, contribuindo para a agilidade e eficácia dos processos de tomada de decisão nas organizações. Este texto também apresenta uma análise comparativa do desempenho das consultas com e sem a utilização de índices, demonstrando o impacto positivo da otimização na prática. Ao final, discute-se acerca de futuros estudos que podem ser feitos no contexto deste trabalho, e de como os resultados obtidos podem contribuir para o desempenho do sistema da empresa em questão.

Palavras-chave:

Otimização de Consultas SQL; Indexação; Desempenho de Consultas; Banco de Dados

1. INTRODUÇÃO

Com o advento de um mundo cada vez mais orientado por dados, a eficiência na recuperação de informações torna-se essencial para as operações do dia a dia. Nesse contexto, a otimização de consultas de banco de dados desempenha um papel crucial, facilitando o acesso rápido e econômico aos dados requisitados pelo usuário. Esta prática visa reduzir o tempo e o custo associados à recuperação de dados, proporcionando ao Sistema de Gerenciamento de Banco de Dados (SGBD) uma rota mais eficiente para o processamento dessas consultas (COUTO, 2006).

Conforme o Barnhill (2019), a otimização de consultas em um banco de dados relacional é um aspecto crítico para melhorar o desempenho do sistema e fornecer respostas rápidas. Uma das estratégias para alcançar isso é por meio da indexação. A indexação acelera as consultas, criando ponteiros para onde os dados estão armazenados em um banco de dados, permitindo que a informação seja recuperada mais rapidamente.

No SQL Server, um índice clusterizado não representa um índice independente, mas sim, estabelece a organização física dos dados na tabela. Essa organização é determinada pela chave do

¹Discente de Bacharelado em Ciência da Computação, IFSULDEMINAS – Campus Passos. E-mail: gregory.fagundes@alunos.ifsuldeminas.edu.br.

²Orientador, IFSULDEMINAS – Campus Passos. E-mail: hiran.ferreira@ifsuldeminas.edu.br.

índice clusterizado, o que define a sequência na qual as linhas de dados são armazenadas fisicamente. Tornando a busca de dados mais eficiente. Devido à natureza dessa ordenação física, é importante notar que cada tabela pode abrigar apenas um único índice clusterizado (JORGENSEN et al., 2012).

Conforme detalhado por Elmasri e Navathe (2011), dentre os diversos aspectos que podem ser ajustados para otimizar o desempenho de um sistema de banco de dados, como a normalização de tabelas, otimização de consultas e particionamento de tabelas, a indexação de colunas se demonstra interessante, pois este método pode fornecer melhorias sem a necessidade de alterar a estrutura pré-existente do banco de dados e consequentemente de sistemas que o usem.

Portanto, este trabalho se concentra na otimização e aprimoramento do desempenho dos resultados de consultas SQL. O foco foi sobre consultas que retornavam um grande volume de dados em um intervalo de tempo específico.

2. MATERIAL E MÉTODOS

Nesta seção são apresentadas as etapas necessárias para o desenvolvimento desse trabalho, onde o foco é voltado para uma análise qualitativa do fenômeno em estudo.

Primeiramente foi feita uma avaliação das tabelas e colunas relevantes para o estudo, com foco particular nas tabelas mais utilizadas pelo sistema, para a inserção e leitura de dados. Por se tratar de um sistema de vendas para postos de combustíveis, foram selecionadas as tabelas tbAbastecimentos e tbVendas, que se mostraram mais significativas. Nas duas tabelas, foram selecionadas as colunas de datas, consideradas fundamentais para a geração de relatórios, tendo em vista que grande maioria dos relatórios são filtrados por períodos recentes de data.

Posteriormente, para monitorar o tempo de execução das consultas, foi utilizado o plano de consulta permitindo a obter o tempo real da execução de uma consulta. Foi anotado o tempo de execução de cada consulta antes da criação dos índices para servir como comparação para o tempo de execução após a criação dos índices. Para garantir resultados precisos, os caches de execução do plano de consulta e do buffer de dados foram limpos antes de cada consulta usando os comandos “DBCC FREEPROCCACHE” e “DBCC DROPCLEANBUFFERS”, respectivamente.

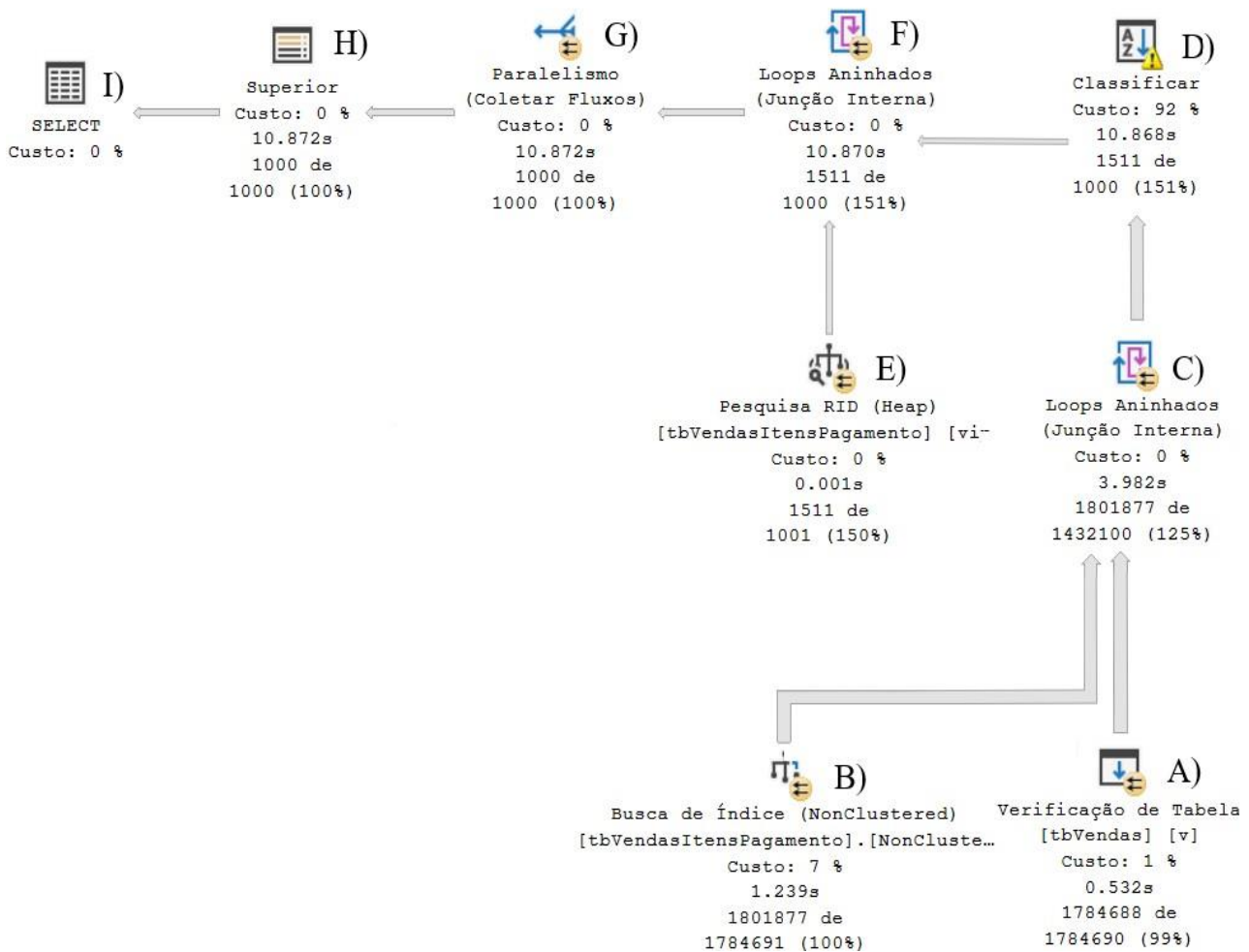
Por fim, os resultados foram documentados em um quadro comparativo, com o tempo de execução antes e após a indexação para cada consulta e a porcentagem de melhoria no tempo de execução que foi calculada e analisada. Permitindo entender melhor o impacto real da criação de índices no desempenho das consultas SQL.

3. RESULTADOS E DISCUSSÃO

Foram realizados cerca de oito testes relacionados ao desempenho de consultas, no entanto, apresentaremos um deles (Figura 1), tendo em vista a disponibilidade máxima de páginas permitidas.

Figura 1 - Plano de execução da primeira análise sem índice referente a tbVendas.

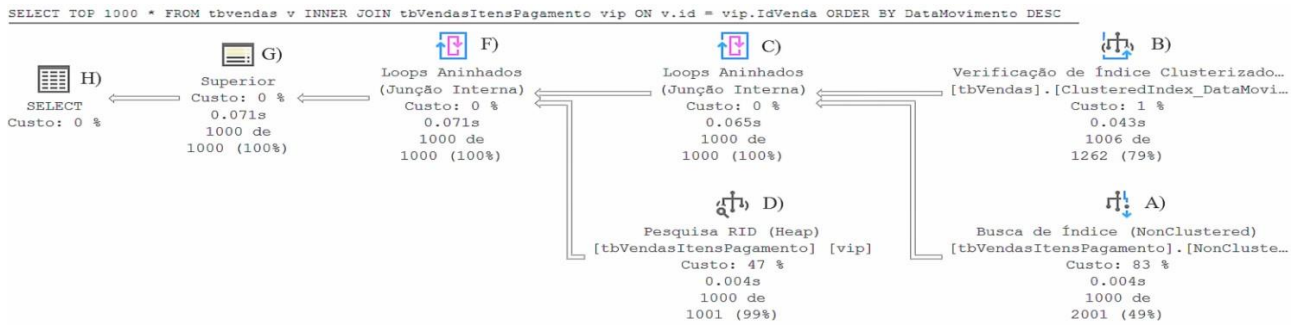
SELECT TOP 1000 * FROM tbvendas v INNER JOIN tbVendasItensPagamento vip ON v.id = vip.IdVenda ORDER BY DataMovimento DESC



Fonte: Autores.

A Figura 1 ilustra o plano de execução de uma consulta envolvendo duas tabelas: as primeiras 1000 vendas ordenadas pela data de venda. Neste plano, o fluxo de execução ocorre da direita para a esquerda. O primeiro passo foi realizar uma varredura completa da tabela de vendas (tbVendas) representado em A) e simultaneamente, em B) é feita uma busca utilizando índices na tabela de itens de pagamento das vendas (tbVendasItensPagamento). Após estas duas operações, as linhas de dados dessas duas tabelas foram combinadas em C) e posteriormente ordenadas em D), e em E) é feito em paralelo uma Pesquisa por id da linha da tabela tbVendasItensPagamento para recuperar as informações das demais colunas sendo feita a junção com o resultado da ordenação em F). Em seguida em G) é aplicado o operador de paralelismo afim de distribuir as entradas em vários núcleos para melhorar o desempenho, e por fim em H) é retornado as 1000 primeiras linhas e exibido em I). É importante notar que a etapa que mais consumiu recursos, ou seja, teve o maior impacto no desempenho da consulta, foi a etapa de ordenação dos dados em D), com um custo de 92%.

Figura 2 - Plano de execução da primeira análise com índice referente a tbVendas.



Fonte: Autores.

Na Figura 2, é apresentada uma abordagem diferente para a mesma consulta realizada na Figura 1. No entanto, foi implementado um índice clusterizado, que organiza a coluna 'dataMovimento' (correspondente à data da venda) em sequência decrescente. Com este índice, a busca se tornou mais direcionada ao utilizar o operador em A), fazendo a leitura de apenas 1000 linhas, comparado com a verificação de mais de 1.780.000 linhas na consulta da Figura 1. Como resultado, menos processos foram necessários, eliminando a necessidade de ordenar os dados posteriormente. Isso levou a uma grande diminuição no tempo de execução.

4. CONCLUSÃO

A análise das tabelas demonstrou a eficiência dos índices na melhoria do tempo de execução das consultas. Quando comparadas as consultas em tabelas sem índices com aquelas realizadas em tabelas com índices, houve uma diminuição notável no tempo de execução, demonstrando a eficácia dos índices para economizar tempo e facilitar as operações da empresa em questão.

Além disso, a otimização do tempo de manipulação de dados diários pode trazer melhorias significativas para as operações da empresa, tornando seus processos mais ágeis e eficazes. Em resumo, este trabalho demonstra que um melhor entendimento e aplicação dos índices pode resultar em ganhos significativos de eficiência e eficácia operacional.

REFERÊNCIAS

BARNHILL, B. **Indexing**. Disponível em: <<https://dataschool.com/sql-optimization/how-indexing-works/>>. Acesso em: 12 de agosto de 2023.

COUTO, Cesar Francisco Moura. **Um arcabouço orientado por aspectos para implementação automatizada de persistência**. 2006. 98f. Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Belo Horizonte, 2006.

ELMASRI, Ramez; NAVATHE, Shamkant B. *Sistema de banco de dados*. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

JORGENSEN, A. *Microsoft SQL Server 2012 bible*. Indianápolis: John Wiley & Sons, 2012.