

## APLICABILIDADE DE TÉCNICAS DE OTIMIZAÇÃO DE CONSULTAS: UM ESTUDO DE CASO

Mayara Kaynne Fragoso Cabral<sup>1</sup>, Vanessa Aires Corado<sup>2</sup>, João Carlos Bruno<sup>3</sup>

<sup>1</sup> Instituto Federal de Ciência e Tecnologia do Tocantins, Porto Nacional – IFTO. e-mail: [mayarakf@ifto.edu.br](mailto:mayarakf@ifto.edu.br)

<sup>2</sup> Graduanda do curso de Licenciatura em Computação do Instituto Federal de Ciência e Tecnologia do Tocantins, Porto Nacional. e-mail: [vanessaa.corado@gmail.com](mailto:vanessaa.corado@gmail.com)

<sup>3</sup> Graduando do curso de Direito na Faculdade Serra do Carmo – Palmas – TO. e-mail: [733@serradocarmo.edu.br](mailto:733@serradocarmo.edu.br)

**Resumo:** As consultas SQL são uma das principais origens de perda de desempenho em bancos de dados, porém há diversas técnicas que podem ser utilizadas para otimizá-las. Com o objetivo de testar a viabilidade de técnicas consideradas de menor impacto e a facilidade de seu uso, foi realizado um estudo de caso em um banco de dados de um Ambiente Virtual de Aprendizagem (AVA). Identificou-se dentre as consultas SQL utilizada pela aplicação aquelas com maior tempo de execução. Aplicou-se nestas consultas quatro técnicas de otimização analisando o novo tempo de resposta da consulta, e a facilidade de aplicação da técnica. Ao término do estudo constatou-se que a utilização de técnicas simples podem influenciar de forma extrema na diminuição do tempo de resposta da consulta, e ainda que uma elaboração ruim na escrita da consulta afeta diretamente o desempenho da aplicação, impactando diretamente no usuário do sistema. Nos testes realizados obteve-se um tempo de resposta em média 30,35 % menor após a aplicação das técnicas de otimização, comprovando a simplicidade e viabilidade de uso destas técnicas.

**Palavras-chave:** Otimização de Consultas, Reescrita de Consultas, Índices.

### 1. INTRODUÇÃO

As aplicações Web estão cada vez mais complexas e tem se expandido rapidamente entre as organizações. Neste tipo de aplicação o tempo de resposta deve ser mínimo visto que está em jogo não somente a paciência do usuário, mas a credibilidade perante o mesmo. Um fator que pode influenciar o desempenho da aplicação é tempo de resposta do banco de dados, por isso é muito comum para os desenvolvedores de softwares começarem a otimização das aplicações, revisando-se as chamadas ao banco de dados. Através do estudo do desempenho e otimização de consultas SQL, pode-se minimizar do tempo de resposta do servidor de banco de dados.

Segundo Cintra e Capelo (2005) “A otimização de consulta é o processo de selecionar o plano de avaliação de consulta mais eficiente para uma consulta”. Os gerenciadores de banco de dados (SGDB) possuem otimizadores de consultas que selecionam o plano de execução mais eficiente para cada consulta. Mas uma consulta SQL mal escrita leva o otimizador a utilizar um caminho que nem sempre é o mais adequado, o que gera um plano de execução que normalmente compromete o desempenho.

O estudo de caso foi realizado na Fundação Universidade do Tocantins, uma instituição de ensino que dentre os cursos oferecidos possuía mais de noventa mil alunos matriculados, somente na modalidade tele presencial. Por isso a aplicação escolhida para estudo de caso foi o AVA (Ambiente Virtual de Aprendizagem) da instituição, por ser uma aplicação web, ter uma grande quantidade de acessos simultâneos e um banco de dados gigantesco com milhões de registros.

Através do estudo realizado nesta aplicação virtual de aprendizagem foi identificado as consultas mais caras e executadas com maior frequência pelos usuários da aplicação, em seguida aplicado técnicas de otimização nestas consultas. Existem inúmeras técnicas de otimização que podem ser usadas para se alcançar desempenho. Assim este artigo tem o objetivo de apontar como é possível aprimorar o tempo de resposta da aplicação simplesmente otimizando os comandos T-SQL, trazendo os resultados obtidos na aplicação destas técnicas de

otimização, e fornecendo suporte aos administradores de banco de dados e desenvolvedores de softwares.

## 2. MATERIAL E MÉTODOS

Para a realização do estudo de caso usou-se o servidor de teste da própria instituição, onde neste estava uma cópia da aplicação e do banco de dados. Neste são testados várias aplicações, por diferentes desenvolvedores da instituição, assim no momento de execução dos testes, o número de processos e conexões simultâneas neste servidor pode influenciar no tempo de processamento e resposta do processador. Por este motivo, a consulta original e a consulta otimizada, a cada técnica de otimização, foram testada imediatamente uma após a outra e sempre no mesmo dia e horário.

Na realização dos testes usou-se seguintes softwares, com as seguintes finalidades:

- SQL Server 2005 que é o gerenciador do banco de dados da aplicação em estudo;
- SQL Profiler, utilizado para monitorar as sessões ativas que estão sendo executadas no banco de dados e identificar as consultas consideradas com pior desempenho;
- ADT( DataBase Engine Tunning Advisor), utilizado para analisar os arquivos de carga de trabalho da aplicação e propor alterações na consulta escolhida para estudo de caso.
- SQL Server Query Analyzer: aplicado as técnicas de otimização, esta ferramenta foi utilizada para obter uma exibição visual do plano de execução do SQL Server e assim podermos comparar o tempo de resposta da consulta antes e depois do uso da técnica

Definidos o ambiente e ferramentas, para a identificação de possíveis causadores de baixo desempenho da aplicação no banco de dados, foram-se realizados os seguintes procedimentos:

**Procedimento 1:** Na ferramenta Profiler SQL criou-se um novo trace, isto é, uma definição de como iremos capturar os dados. O Profiler disponibiliza vários modelos de trace onde cada um faz a captura de determinados eventos, podendo filtrar as atividades capturada através da especificação de filtros de forma que sejam exibidas apenas as consultas executadas no banco de dados especificado. Para o teste foi criado um trace utilizando o modelo TSQL\_Duration.tdf, que coleta informações sobre todos os comandos T-SQL que foram enviados para o servidor SQL Server e o tempo de duração para a execução dos comandos. Foi aplicado no modelo filtros com o nome do banco a ser monitorado, e o tipo de texto que almejava-se capturar, como mostrado abaixo:

- DataBaseName - Like 'Db\_Central'
- TextData - Like 'Select%' - NotLike 'Delete%', 'Update%', 'Insert%'

**Procedimento 2:** Gerou-se atividade no banco de dados a partir da navegação nas interfaces da aplicação.

**Procedimento 3:** Gerado consultas no banco, voltou-se ao Profiler, e parou-se a captura e salvou-se os eventos capturados em arquivo. A partir deste arquivo foi analisado as consultas capturadas, identificando as que apresentaram maior tempo para serem concluída. Dentre estas foi escolhida a que teve o tempo de execução mais elevado para ser utilizada nos testes. A consulta selecionada apresentava duração de 9534 milissegundos, isto é, quase 10 segundos no momento em que foi executado.

A consulta escolhida no estudo de caso é utilizada no Relatório de Log de Navegação do AVA. Este relatório verifica em determinada data quais ferramentas da aplicação cada aluno acessou, com o objetivo de monitorar os acessos dos mesmos dentro da aplicação, podendo-se identificar, por exemplo, a alegação de um aluno não ter postado o trabalho por não conseguir acessar a página dentro da ferramenta, o que acontece com muita frequência por ser a aplicação o principal meio de comunicação do aluno com os professores.

Na consulta são retornadas informações como: nome do aluno, página que ele navegou dentro da aplicação, nome da aplicação, data e horário da navegação, email do aluno, o IP do computador do aluno e a string de navegação.

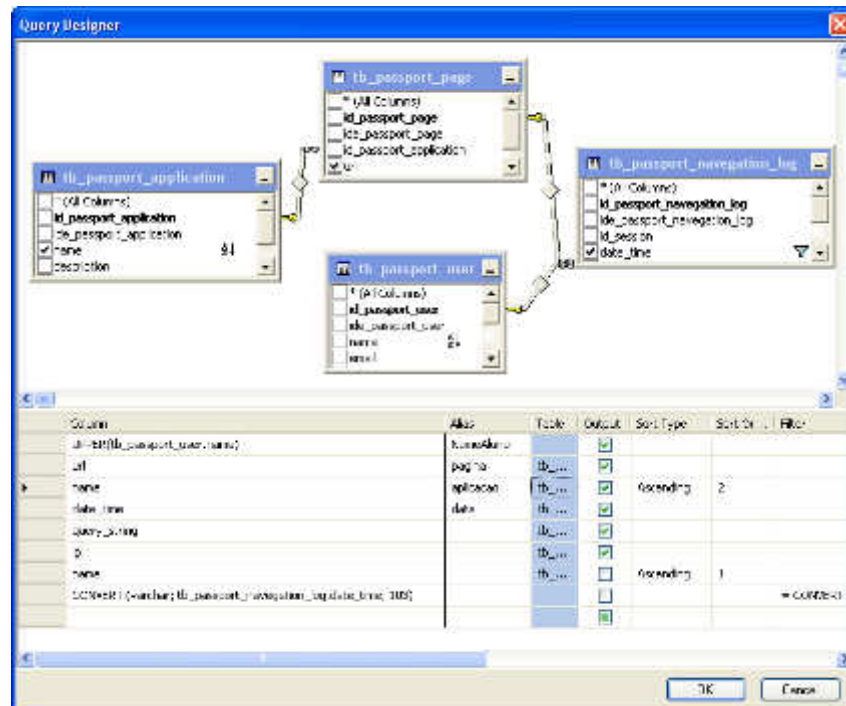


Figura 1: Estrutura da consulta escolhida para o estudo de caso

### 3. RESULTADOS E DISCUSSÃO

Silberschatz (2006) afirma que a otimização de consulta como um o processo de seleção do plano de avaliação de consulta são mais eficiente dentre as muitas estratégias normalmente possíveis para o processamento de determinada consulta. Segundo o autor a diferença no custo entre uma boa estratégia e uma estratégia ruim normalmente é substancial e pode chegar a várias ordens de grandeza. Logo, “vale a pena para o sistema gastar uma quantidade de tempo substancial na seleção de uma boa estratégia para processar uma consulta.”

Em muitos casos especificar uma sentença SQL correta não é o bastante para se conseguir um tempo de resposta mínimo, é necessário indicar ao otimizador deve ser feito. Existe várias maneiras de se otimizar uma consulta, algumas mais simples outras mais complexas.

Neste estudo de caso é apresentado algumas técnicas, dentre as várias disponíveis foi escolhidas as consideradas de menor impacto, isto é, as que gerariam mudanças mínimas no código da aplicação. Para cada técnica testada é explanado sobre seu funcionamento e analisado a aplicabilidade destas no melhoramento do tempo de resposta da consulta escolhida.

#### 3.1 Técnica de Sintonia de Índices

Habitualmente, os índices são utilizados para melhorar o desempenho dos bancos de dados. Um índice permite ao servidor de banco de dados encontrar e trazer linhas específicas com muito mais rapidez do que faria sem o índice. Segundo Battisti (2005) a criação de índices é uma ferramenta poderosa para o projetista do banco de dados, pois o papel principal do índice é reduzir o número de operações de I/O(leitura e escrita(Input/Output) no disco)necessárias para localizar os dados solicitados pela consulta.A identificação de quais campos da tabela são mais indicados para se tornarem índices ou quais já existentes podem ser excluídos, é um

questionamento comum entre os administradores de banco e solução destes problemas muitas das vezes requer conhecimentos avançados.

Para auxiliar na determinação de como implementar os índices, para aumento de performance, foi utilizada a ferramenta DTA (Database Engine Tuning Advisor) disponível no SQL Server 2005. O DTA realiza a análise de performance não apenas a partir de consultas, pode-se analisar todo o banco de dados, e até mesmo a trace gerada no Profile com todos os eventos capturados. A ferramenta possui opções avançadas, possibilitando além da visualização das melhorias, a execução das ações recomendadas pela própria ferramenta. Analisada assim a consulta pela ferramenta, foi recomendado a criação de estatísticas nos campos da tabela tb\_passport\_navegation\_log, que é a principal tabela da consulta e a responsável em armazenar os logs de navegação, e a criação de um índice na mesma tabela para o campo data\_time.

Segundo Silberschatz (2006) a informação estatística sobre as relações é particularmente útil quando vários índices estão disponíveis para ajudar no processamento de uma consulta. A presença dessas estruturas possui uma influência significativa sobre a escolha de uma estratégia de processamento de consulta.

Quando existe a necessidade de se criar um novo índice, uma boa prática é identificar a coluna mais seletiva da tabela, isto é, a coluna com menor repetição de valores. Pois quando é executada uma consulta o otimizador do Banco de Dados verifica se existe um índice para cada cláusula, avalia a utilidade do índice existente, e determina a seletividade da cláusula estimando o custo para encontrar as linhas qualificadas.

A seletividade é a relação estabelecida entre a quantidade de linhas de uma tabela retornadas por uma consulta com a quantidade total de linhas da mesma tabela.

$$\text{Seletividade} = \frac{\text{N}^\circ \text{ de registros retornados}}{\text{N}^\circ \text{ Total de registros}}$$

O índice será útil quando a relação da seletividade é de 5% ou menos. Caso o índice tenha uma seletividade maior de 5%, provavelmente não será utilizado, ou ainda outro índice será escolhido ou a tabela será percorrida através de uma varredura completa nos dados.

Assim dando continuidade aos testes, para identificar a seletividade do campo recomendado pela ferramenta DTA para ser alterado para índice, foi executada a consulta abaixo:

Tabela 1: Teste de seletividade do campo

	<b>Resultado Obtido</b>		
TESTE 1	9534	6635	30,41 %
TESTE 2	37733	5518	85,37 %

Analisando-se o valor retornado para a seletividade da coluna, foi constatado que a criação do índice neste campo realmente era indicada e, portanto foi considerada a recomendação feita pelo DTA, de forma que foi dado continuidade nos procedimentos começados anteriormente.

Foi feito um backup do banco de dados para que se pudesse comparar os tempos de respostas antes e depois de ser executado sem alterações as recomendações pela ferramenta DTA. Comparado o tempo de resposta da consulta no banco de backup com o tempo de resposta do banco original, foi constatado que com a criação do índice e das estatísticas, a consulta diminuiu uma média de 58 % seu tempo de duração, como pode ser visto na Tabela abaixo.

Tabela 2: Teste de sintonia de índices

Testes Realizados	Tempo de Resposta (milissegundos)		% Minimização de resposta
	Backup do Banco	Banco Otimizado	
TESTE 1	9534	6635	30,41 %
TESTE 2	37733	5518	85,37 %

### 3.2 Técnica de Ordenação das Tabelas da Consulta

Outra técnica que pode ser usada na otimização é a de reescrever a consulta. Considerada a de menor impacto, visto que os ajustes feitos na consulta afetam apenas a própria consulta não se propagando para outras aplicações que acessam as tabelas envolvidas na instrução SQL.

Para testar a aplicabilidade de cada critério usado na técnica de reescrita da consulta foi utilizado o banco já com a técnica de sintonia de índices aplicada.

As ordenações das tabelas dentro da consulta podem influenciar no tempo de resposta da mesma. Por isso de acordo a ordem em que estas são colocadas, o gerenciador do banco terá que percorrer um número maior ou menor de registro, como foi identificado nos testes realizados:

Tabela 3: Resultado obtidos com a técnica de ordenação das tabelas

Ordem	Organização das tabelas dentro da consulta	Tempo de Resposta	
		Teste 1	Teste 2
Ordenação 1	Ordem original	9534	6967
Ordenação 2	Mudança na ordenação 1	10770	7397
Ordenação 3	Mudança na ordenação 2	<b>6400</b>	<b>6398</b>

Como pode-se observar na tabela acima a medida que mudou-se a organização das tabelas dentro da consulta, foi-se obtendo tempos de resposta em milissegundos diferentes. Isto comprova que a ordem da tabela influencia no resultado final. No teste realizado na consulta problemática, obteve-se um ganho de 20,5 % somente reordenando as tabelas da consulta.

### 3.3 Uso de Comandos que produzem sort

Segundo COUTO (2015), os comandos que produzem sort (group by, order by, distinct, etc.) devem ser usados somente quando necessário, pois no sort, o banco recupera todas as informações necessárias primeiro e depois as ordena.

Na consulta em estudo, usava-se o comando de ordenação Order by, para ordenar os registros pelo nome do aluno e pelo nome da aplicação. Com o objetivo de verificar se este comando poderia estar mesmo alterando o tempo de resposta da consulta, foram realizados os testes abaixo:

Tabela 4: Resultados obtidos o uso de comandos de ordenação dos campos da tabela

Testes Realizados	Tempo de Resposta (milissegundos)		% Tempo de Minimização de resposta
	Usando o comando Order By	Sem o comando Order By	
TESTE 1	6635 ms	4582 ms	30,95 %
TESTE 2	6443 ms	4437 ms	31,13 %

Com a aplicação desta técnica foi confirmado que o uso de comando de ordenação aumenta o tempo de resposta da consulta. Nos testes o tempo de duração diminuiu uma média de 31 % se não utilizado o comando order by.

### 3.4 Junção de Tabelas

Em muitas discussões há certa confusão se existe alguma diferença de performance ao utilizar o operadores Where ou Inner Join nos relacionamentos das tabelas.

Segundo o site da Microsoft SQL Server às vezes, você pode reescrever uma subconsulta para usar JOIN e obter melhor desempenho. A vantagem de criar uma subconsulta JOIN é poder avaliar as tabelas em uma ordem diferente daquela definida pela consulta. A vantagem de usar uma subconsulta é que geralmente não é necessário examinar todas as linhas da subconsulta para avaliar a expressão.

Assim com o objetivo de responder a estes questionamentos, e identificar qual operador deve ser utilizado quando se deseja otimizar uma consulta, foi realizado os testes utilizando-se ambos os operadores, obtendo-se o resultado que pode ser visto abaixo:

Tabela 5: Comparação entre os operadores Where e Inner Join

Testes Realizados	Tempo de Resposta		Diferença no Tempo de Resposta(milissegundos)	Ordem de execução
	INNER JOIN	WHERE		
TESTE 1	6255 ms	6595 ms	340 ms	Where, Inner Join
TESTE 2	9499 ms	5701ms	3798 ms	Inner Join, Where

Nos testes realizados observou-se que sempre a primeira consulta executada apresentava um tempo maior, como pode ser visto na figura 3 que mostra o primeiro teste, e na figura 4 que mostra o segundo teste. Com esta constatação, concluiu-se que ambos os comando de junção são vistos como iguais pelo SQL Server, pois a medida que a primeira consulta foi executada, o SQL Server guardou em memória as páginas de dados utilizadas para atender a consulta bem como os planos de execução utilizados. Quando houver uma nova requisição idêntica a anterior, o SQL Server já tinha o plano de execução e as páginas de dados e executou-a bem mais rapidamente.

Realizado a aplicação das técnicas escolhida, obteve-se os seguintes resultados em relação ao tempo de resposta da consulta:

Tabela 6: Comparação dos resultados obtidos em cada técnica de otimização

Técnica	% Tempo de Minimização de resposta
Sintonizar os índices	58%
Mudar ordem das tabelas	20,5 %

Não utilização do <i>Order by</i>	31 %
Utilização do Inner Join ao invés do Where	0%

Desta forma após a aplicação destas quatro técnicas obteve-se uma consulta otimizada com o seguinte formato:

Tabela 7: Consulta otimizada

```
SELECT  tb_passport_user.name AS NomeAluno,
        tb_passport_page.url as pagina,
        tb_passport_application.name as aplicacao,
        tb_passport_navegation_log.date_time as data,
        tb_passport_navegation_log.query_string,
        tb_passport_navegation_log.ip
FROM    tb_passport_user INNER JOIN
tb_passport_navegation_log          ON          tb_passport_user.id_passport_user          =
tb_passport_navegation_log.id_passport_user  INNER JOIN          tb_passport_page          ON
tb_passport_navegation_log.id_passport_page = tb_passport_page.id_passport_page INNER JOIN
tb_passport_application          ON          tb_passport_page.id_passport_application      =
tb_passport_application.id_passport_application
WHERE   (CONVERT(varchar, tb_passport_navegation_log.date_time, 103) =
CONVERT(varchar, '21/11/2008', 103))
```

Ao final da aplicação das técnicas de otimização, não houve mudanças drásticas na forma como a consulta havia sido construída, porém o tempo que a consulta levava para retornar os dados diminuiu em média 30,35 %, isto é, um pouco mais de dois segundo, como pode ser visto na tabela 8 abaixo.

Tabela 8: Resultados obtidos antes e após a aplicação das técnicas de otimização de consultas

Testes Realizados	Tempo de Resposta		Diferença no Tempo de Resposta	Tempo de Minimização de resposta	
	Consulta original	Consulta otimizada		Em %	Em segundos
TESTE 1	5876	4270	1606	27,3 %	1,76 s
TESTE 2	6868	4568	2300	33,4%	2,29 s

Considerando-se que foi reduzido o tempo de execução de uma única consulta de média de 6,3 para 4,3 segundos, 2,0 segundos pode não ser um tempo tão relevante se a consulta não for executada com frequência durante o dia. No entanto, se a consulta for executada com muita frequência por um número alto de usuários simultâneos, a melhoria de desempenho pode ser muito eficiente.

## 6. CONCLUSÕES

Com todos os testes realizados foi constatado que a maneira como a consulta é construída influencia de forma extrema o tempo de resposta da aplicação, quando mal elaborada, como no caso da consulta usada nos testes, o desempenho da aplicação é afetado, impactando diretamente no tempo de espera do usuário pela resposta da aplicação.

Concluiu-se que há muitas outras formas de se otimizar uma consulta, mas dentre as técnicas apresentadas e testadas, todas mostraram-se altamente aplicáveis. O uso de cada uma

delas mostrou-se também muito simples, e que podem ser aplicadas sem que haja a necessidade de alteração no código da aplicação, isto é, técnicas realmente de baixo impacto.

O uso destas técnicas mostrou-se como ótimas práticas, e devem ser por todos os administradores de banco de dados (DBA) e desenvolvedores de softwares. A recomendação é que sejam adotadas preferencialmente antes da construção das consultas, mas se esta ordem não for seguida, o estudo de caso mostrou também que como é possível se identificar as consultas de baixo desempenho utilizadas pela aplicação, e como e quais técnicas se utilizar para gerar melhores resultados em relação do tempo de resposta das consultas e do banco de dados em geral.

Assim os resultados obtidos com os testes foram muito positivos, porém para visualizar realmente o impacto dessas mudanças, se o custo e, conseqüentemente, o tempo de execução realmente diminuiu, seria necessária a reprodução das mudanças no ambiente real da consulta analisada, visto que os testes foram realizados no servidor de teste da instituição e não no servidor que realmente hospeda a aplicação.

## REFERÊNCIAS

BATTISTI, Julio. **SQL Server 2005 Administração e Desenvolvimento Curso Completo**. Axcel Books do Brasil Editora. ISBN 85-7323-249-8. 2005

CINTRA, J. P. S.; CAPELLO, R. **Otimização de consultas relacionais**. Universidade Estadual de Campinas. Trabalho de Pós-graduação em Ciência da Computação. 2005. Disponível em <http://www.ic.unicamp.br/~geovane/mo410-091/Ch15-Resumo.pdf> Acesso em 06 out 2015

COUTO, Eder. **Aumentando a performance da aplicação através da otimização de SQL**. Disponível em: <[Aumentando a performance da aplicação através da otimização de SQL](http://www.devmedia.com.br/aumentando-a-performance-da-aplicacao-atraves-da-otimizacao-de-sql/3229#ixzz3nnw9A6vq)  
<http://www.devmedia.com.br/aumentando-a-performance-da-aplicacao-atraves-da-otimizacao-de-sql/3229#ixzz3nnw9A6vq>>. Acesso em 06 out 2015

MICROSOFT SQL SERVER. Ajuste do desempenho de consulta (SQL Server Compact). Disponível em: < <https://technet.microsoft.com/pt-br/library/ms172984%28v=sql.110%29.aspx#feedback>. Acesso em 06 out 2015

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de banco de dados**. 5. ed. São Paulo: Elsevier Editora Ltda, 2006.