

**UNIVERSIDADE ESTÁCIO DE SÁ
FACULDADE INTEGRADA DO CEARÁ**



**CURSO DE ESPECIALIZAÇÃO LATO SENSU DE
GESTÃO DE BANCO DE DADOS**

ANTONIO ALEX DE SOUZA

**MELHORIA DE DESEMPENHO DE BANCO DE DADOS
UTILIZANDO O MICROSOFT SQL SERVER**

FORTALEZA (CE)

2016

**MELHORIA DE DESEMPENHO DE BANCO DE DADOS
UTILIZANDO O MICROSOFT SQL SERVER**

Artigo científico apresentado ao curso de Pós-graduação de Gestão de Banco de Dados pela Faculdade Integrada do Ceará, como requisito para a obtenção do grau de Especialista.

Orientador: Eliseu Castelo Branco Júnior

FORTALEZA (CE)

2016

MELHORIA DE DESEMPENHO DE BANCO DE DADOS UTILIZANDO O MICROSOFT SQL SERVER

Antonio Alex de Souza¹

Eliseu Castelo Branco Júnior²

RESUMO

Neste artigo, demonstra-se a aplicação de boas práticas relacionadas a desempenho de banco de dados com o objetivo de alcançar os melhores tempos de resposta para as requisições. Além da fundamentação teórica e técnicas utilizadas para melhoria de desempenho de banco de dados, foi realizado um estudo de caso para validar as técnicas definidas e posteriormente uma análise dos resultados. Contribuindo para minimizar o tempo de resposta às requisições aos bancos de dados, foram utilizadas as técnicas de refatoração, indexação, fragmentação de índices e particionamento de tabelas.

Palavras-Chave: Banco de Dados. Performance. Melhores Práticas.

ABSTRACT

This paper demonstrates the application of good practices related to performance database in order to achieve the best response times for requests. Besides the presentation of theoretical basis and techniques used to improve database performance, we performed a case study to validate the techniques and then set an analysis of the results. Contributing to minimize the response time to requests to databases, we used the techniques of refactoring, indexing, index fragmentation and partitioning of tables.

Keywords: Database. Performance. Best practices.

¹ Aluno do curso de MBA em Banco de Dados do Centro Universitário Estácio do Ceará.

² Professor Mestre orientador do artigo.

1. INTRODUÇÃO

Para controlar este imenso volume de dados, que circulam atualmente, é necessário um sistema gerenciador de banco de dados (SGBD) que lhe proporcione este controle com uma maior confiabilidade, segurança, robustez e desempenho, cujos fatores podem determinar o sucesso ou o fracasso da empresa.

Com a crescente necessidade de armazenamento de dados, os bancos de dados apresentam-se cada vez maiores e com um volume transacional muito alto. Junto com este crescimento foram surgindo algumas técnicas que melhoram a forma de armazenar e recuperar estes dados e informações, as quais são abordadas neste trabalho.

A escolha do tema é justificada principalmente pela crescente necessidade das empresas em obter o máximo de informações no menor tempo possível, em que se utilizam técnicas e boas práticas com o objetivo de melhorar o desempenho dos bancos de dados.

Este trabalho tem como principal objetivo estudar técnicas e as melhores práticas utilizadas no mercado atualmente para melhoria de desempenho em banco de dados.

Os objetivos específicos visam: a) apresentar uma fundamentação teórica das técnicas utilizadas; b) pesquisar e tabular dados sobre a utilização de boas práticas de desempenho de banco de dados, com base em um cenário real; c) apresentar o resultado da pesquisa.

Seguindo-se a esta introdução, tem-se a metodologia da pesquisa em sua classificação e tipologia.

Em seguida, apresenta-se banco de dado em seus conceitos, índices, particionamento de tabelas, refatoração de banco de dados dentre outras técnicas de melhoria de desempenho.

A sessão seguinte apresenta um estudo de caso, onde serão aplicadas algumas das melhores práticas de melhoria de desempenho em banco de dados e logo em seguida, analisamos os dados obtidos com o uso de cada uma das praticas aplicadas.

2. METODOLOGIA DE PESQUISA

A metodologia utilizada em uma pesquisa traduz a forma como foi desenvolvida. No caso específico deste estudo, quanto aos meios ou aos procedimentos técnicos, foi bibliográfica cujos principais assuntos abordados foram banco de dados e melhores práticas para obter maior desempenho em consultas a banco de dados. Exemplificando esse tipo de pesquisa tem-se que:

[...] não deve ser confundida, como acontece frequentemente, com a pesquisa de documentos. O levantamento bibliográfico é mais amplo do que a pesquisa documental, embora possa ser realizado simultaneamente com a pesquisa de campo e de laboratório. A pesquisa bibliográfica tem por finalidade conhecer as diferentes formas de contribuição científica que se realizaram sobre determinado assunto ou fenômeno. Normalmente, o levantamento bibliográfico é realizado em bibliotecas públicas, faculdades, universidades e, especialmente, naqueles acervos que fazem parte do catálogo coletivo e das bibliotecas virtuais” (OLIVEIRA, 2010, p. 119).

Bibliográfica, pois, segundo Vergara (2010, p. 43), foi realizada a partir de material teórico já existente em livros, revistas, jornais, artigos e sites da Internet, utilizando dados secundários, ou seja, material já publicado anteriormente.

Do ponto de vista de seus objetivos, a pesquisa pode ser considerada descritiva e exploratória. Descritiva, por descrever os principais aspectos inerentes a banco de dados e melhores práticas em desempenho de consultas. Também exploratória por explorar casos verídicos de utilização dessas melhores práticas de desempenho em consultas aplicada a uma empresa real do ramo metalúrgico.

Trata-se de uma pesquisa quantitativa, esclarece Fonseca (2002, p. 20), diferentemente da pesquisa qualitativa, os resultados da pesquisa quantitativa podem ser quantificados. Como as amostras geralmente são grandes e consideradas representativas da população, os resultados são tomados como se constituíssem um retrato real de toda a população alvo da pesquisa. A pesquisa quantitativa se centra na objetividade. Influenciada pelo positivismo, considera que a realidade só pode ser compreendida com base na análise de dados brutos, recolhidos com o auxílio de instrumentos padronizados e neutros. A pesquisa quantitativa recorre à linguagem matemática para descrever as causas de um fenômeno, as relações entre variáveis, etc.

3. FUNDAMENTAÇÃO TEÓRICA

3.1 Banco de Dados

Conforme Silberschatz et al. (2012), um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O conjunto de dados, comumente chamado de banco de dados, contém informações sobre uma empresa em particular. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados.

Sistemas de banco de dados são projetados para gerir grandes volumes de informações. O gerenciamento de informação implica a definição das estruturas de armazenamento das informações e a definição dos mecanismos para a manipulação dessas informações. Ainda, um sistema de banco de dados deve garantir a segurança das informações armazenadas contra eventuais problemas com o sistema, além de impedir tentativas de acesso não autorizadas. Se os dados são compartilhados por diversos usuários, o sistema deve evitar a ocorrência de resultados anômalos.

Um banco de dados (sua abreviatura é BD, em inglês DB, database) é uma entidade na qual é possível armazenar dados de maneira estruturada e com a menor redundância possível. Estes dados devem poder ser utilizadas por programas, por usuários diferentes. (CCM(1))

Esses bancos de dados permitem aos usuários consultarem diretamente seus dados e com base nesses dados, sejam eles relacionados ou não, gerarem informações valiosas às empresas.

3.2 Desempenho de Banco de Dados

A palavra Desempenho têm origem de "des", negativo, e "empenho", ou seja, "Tirar do penhor, resgatar", e "Penhor" vêm do Latim jurídico PIGNUS, que significa "Garantia dada pelo Devedor ao Credor". No dicionário significa "Cumprimento de uma obrigação". Qualidade da representação ou interpretação de um artista". Interessante, você não acha?

De qualquer maneira, as definições nos remetem a idéias muito limitadas, ações arbitrárias e de mera troca de favores. Tudo isto soa como o mínimo que devemos fazer para atingir um determinado resultado. Hoje, o desempenho virou uma

espécie de "unidade de medida". Ele está presente quando nos dizem que o Desempenho daquele computador é superior ao outro, que o desempenho deste atleta ficou abaixo do esperado e que o desempenho do colaborador Fulano Beltrano, em sua última "Avaliação de Desempenho", demonstrou que está apto para determinada função. (CURBETE, online)

Ao projetar um banco de dados, deve certificar-se de que o banco de dados executa todas as funções importantes correta e rapidamente. Alguns problemas de desempenho podem ser resolvidos depois que o banco de dados estiver em produção. Entretanto, outros problemas de desempenho podem ser resultado de um mau projeto de banco de dados, só podendo ser resolvidos pela alteração da estrutura e projeto do banco de dados. (MICROSOFT (2)).

Nos próximos tópicos, são definidas algumas das melhores práticas para melhoria de desempenho em um banco de dados, que se utiliza neste trabalho.

3.2.1 Organização de arquivos de Banco de Dados

Silberschatz et al. (2012) afirmam que para reduzir o tempo de acesso aos blocos, eles podem ser organizados em um disco de forma a corresponderem aproximadamente à maneira pela qual se espera que os dados sejam acessados. Por exemplo, se se almeja que um arquivo seja acessado sequencialmente, então o ideal seria manter todos os blocos de um arquivo sequencialmente em cilindros adjacentes.

Outra abordagem para reduzir a latência de escritas é a utilização de um disco de log - ou seja, um disco voltado para armazenar um log sequencial - muito similar ao buffer de RAM não volátil. Todo acesso ao disco de log é sequencial, eliminando essencialmente o tempo de procura, e diversos blocos consecutivos podem ser escritos simultaneamente, tornando a escrita em discos de log diversas vezes mais rápida que escritas aleatórias.

Assim como antes, os dados também têm de ser escritos em suas localizações reais no disco, mas essa operação de escrita pode ser feita sem que o sistema de banco de dados tenha de esperar para que ele seja completado. Após uma falha de sistema, o sistema lê o disco de log para encontrar aquelas operações de escrita que não tenham sido completadas e então completá-las (SILBERSCHATZ et al., 2012).

3.2.2 Refatoração de Banco de Dados

Refactoring de Banco de Dados não é mágica! Consiste em simples mudanças no esquema do banco de dados (tabelas, views, trigger, procedures etc.) que melhoram o design sem alterar a semântica. Em outras palavras, não se pode adicionar novas funcionalidades ou interromper as funcionalidades existentes. Também não é possível adicionar novos dados nem mudar o significado dos que já estão persistidos. O processo de refactoring do banco de dados define a forma de evoluir de maneira segura um esquema de banco de dados em pequenos passos (incremental e evolutivamente).

Martin Fowler define refactoring da seguinte forma: “É um processo de alteração em um sistema de software, de modo que, o comportamento externo do código não mude, mas que sua estrutura interna seja melhorada.” (DEVMEDIA, online).

3.2.3 Índices

Um índice para um arquivo do sistema trabalha de forma parecida com um catálogo para um livro em uma biblioteca. Se estiver procurando um livro de um autor em particular, procura-se no catálogo de autor e um cartão que informará onde encontrar o livro. Para ajudar na procura no catálogo, a biblioteca mantém os cartões em ordem alfabética, de forma que não se tenha de verificar cada um dos cartões para encontrar o que se procura.

Em banco de dados do mundo real, os índices do tipo descrito acima podem ser muito grandes para serem manipulados eficientemente. Em vez disso, técnicas mais sofisticadas de índices podem ser utilizadas. Há dois tipos básicos de índices:

- Índices ordenados. Esses índices baseiam-se na ordenação dos valores
- Índices hash. Esses índices baseiam-se na distribuição uniforme dos valores por meio de uma faixa de buckets. O bucket ao qual um valor é atribuído é determinado por uma função, chamada de função hash (SILBERSCHATZ et al., 2012)

A estrutura de índice árvore-B+ é a mais utilizada das diversas estruturas de índices que mantêm sua eficiência, a despeito da inserção ou remoção de dados. Um índice árvore-B+ tem a forma de uma árvore balanceada na qual todos os caminhos a partir da raiz da árvore até uma de suas folhas são do mesmo comprimento. Cada nó

não-folha da árvore possui entre $\lceil n/2 \rceil$ e n filhos, em que n é fixo para uma árvore em particular (SILBERSCHATZ, 2012).

3.2.3.1 Fragmentação de Índices

A fragmentação ocorre quando os índices têm páginas nas quais a ordem lógica, com base no valor de chave, não corresponde à ordem física do arquivo de dados. Índices com fragmentação pesada podem degradar o desempenho da consulta e causar lentidão de resposta do aplicativo (MICROSOFT (1)).

Essa situação ocorre devido ao alto volume de operações no banco de dados, tais como: escrita, leitura, atualização e deleção, que segundo Microsoft (1), no decorrer do tempo, essas modificações podem fazer com que as informações do índice sejam dispersas pelo banco de dados (fragmentadas).

3.2.4 Particionamento de tabela em banco de dados

Conforme Silberschatz et al. (2012), em sua forma mais simples, o paralelismo de I/O refere-se à redução do tempo necessário para recuperar relações do disco por meio do particionamento dessas relações em múltiplos discos. A forma mais comum de particionamento de dados em um ambiente de banco de dados é o particionamento horizontal. No particionamento horizontal, as tuplas de uma relação são divididas (ou desagrupadas) entre muitos discos, tal que cada tupla resida em um disco diferente. Existem diversas técnicas de particionamento, tais como: round-robin, particionamento hash e particionamento por faixa.

Descreve-se melhor o particionamento por faixa, pois é o tipo de particionamento utilizado no estudo. Segundo Silberschatz et al. (2012), esta estratégia distribui faixas contíguas do valor de um atributo para cada disco. Um atributo de particionamento A é escolhido como um vetor de particionamento. Seja a sequência $[v_0, v_1, \dots, v_{n-2}]$ denotando o vetor de particionamento, tal que, se $i < j$, então $v_i < v_j$. A relação é particionada como segue. Considere uma tupla t , tal que $t[A] = x$. Se $x < v_0$, então t é colocada no disco D_0 . Se $x \geq v_{n-2}$, então t é colocada no disco D_{n-1} . Se $v_i \leq x < v_{i+1}$, então t é colocada no disco D_{i+1} .

4. ESTUDO DE CASO

A empresa estudada é uma grande empresa do ramo metalúrgico do estado do Ceará, que está passando por uma determinada situação que gera um grande desconforto em relação ao desempenho de suas consultas.

Utilizam um software de gestão empresarial (ERP), onde uma de suas principais funcionalidades é armazenar as informações do ciclo de produção de seus produtos.

Utilizam mais três outros softwares de calibração, que armazenam informações dos "erros" de calibração de cada um dos produtos, ou seja, durante o ciclo de produção, o produto passa por diversos testes de calibração, exemplo: teste de vazão, volume e fadiga.

4.1 Estrutura atual das tabelas dos Bancos de Dados

A principal tabela do banco de dados do sistema ERP armazena informações tais como: ordem de fabricação, número de série, de lacre, selo do Inmetro, matrícula do operador, entre outras informações de cada um dos produtos e a fase do ciclo que se encontra.

Os bancos de dados dos sistemas de calibração, que apesar de serem de um único fornecedor, apresentam estrutura de tabelas distintas devido à diferença entre versões do software. Diante desta situação, os dados são acessados através de visões (Views) criadas no banco de dados do sistema ERP, para cada um dos bancos de dados de calibração, que trazem somente as informações necessárias para integração com o banco de dados ERP. Estas visões armazenam informações de erros de calibração e o número de série de cada produto.

O relacionamento entre a tabela do ERP e as visões para os bancos de dados de calibração se dá pelo número de série do produto.

4.2 Situação atual

Todos os meses são gerados diversos tipos de relatórios, sejam eles internos ou para órgãos fiscalizadores, que retornam informações dos produtos com seus dados de

calibração e este é o motivo da integração do banco de dados do ERP com os bancos de dados dos sistemas de calibração.

A integração foi realizada com sucesso, mas com o passar do tempo, os usuários começaram a alegar lentidão na geração das consultas que envolvem os bancos de dados integrados, como também perda de desempenho nos sistemas transacionais de uma forma geral quando estas consultas estão sendo executadas, principalmente em aplicações da área produtiva.

4.3 Cenários

Em uma primeira análise foram detectados alguns pontos que poderiam ser revistos objetivando ganho de desempenho, entre estes se podem destacar:

- Estrutura física dos arquivos mal planejada, para ambos os bancos de dados;
- Bancos de dados distintos para um mesmo software (único fornecedor);
- Falta de índices, principalmente para as tabelas do banco de dados de calibração;
- Índices fragmentados na tabela de leituras do sistema ERP.

Diante destes pontos acima descritos, montaram-se alguns cenários utilizando técnicas para melhoria de desempenho. Inicia-se comentando sobre a organização dos arquivos que anteriormente eram armazenados em um mesmo local (arquivos de sistema, de dados, de log etc.) e com a reestruturação, os arquivos de sistema ficaram armazenados em um determinado disco rígido (mesmo disco do sistema operacional), os arquivos de dados e de log ficaram armazenados em outros dois discos rígidos distintos, conforme recomendação do fabricante do sistema gerenciador de banco de dados utilizado que diz: “Por padrão, os dados e logs de transação são colocados na mesma unidade e caminho. Isto é feito para controlar os sistemas de um único disco. Porém, isso não é o ideal para ambientes de produção. Recomenda-se que se coloquem os dados e arquivos de log em discos separados” (MICROSOFT (3)).

Os cenários 1, 2 e 3 foram montados seguindo este esquema de armazenamento, já o cenário 0 foi montado conforme o estado atual.

4.3.1 Cenário 0

O cenário 0 foi montado respeitando a estrutura, os bancos de dados e os recursos utilizados no estado atual (ambiente de produção) e servirá como base de comparação de resultado com os demais cenários que serão montados.

4.3.2 Cenário 1

No cenário 1, utilizou-se a técnica de desfragmentação de índices, em que foram desfragmentados todos os índices das tabelas de leituras do banco de dados do ERP, são elas: TbLds e TbLdsPar. Pode-se ver o nível de fragmentação de um índice, utilizando a visão de sistema: *sys.dm_db_index_physical_stats* do Microsoft SQL Server 2008 R2 (MSSQL) analisando o campo: *avg_fragmentation_in_percent* que indica se seus valores estiverem entre 5% e 30% o ideal seria ser feito uma reorganização (REORGANIZE) do índice e se caso estiver maior que 30% uma recriação (REBUILD) do índice seria a melhor escolha, valores fora destes intervalos são considerados normais.

No caso, usou-se o REBUILD, pois os percentuais de fragmentação foram acima de 30%. A sintaxe do comando é a seguinte:
ALTER INDEX <nome do índice> ON <nome da tabela> REBUILD;

Cenário 1 Tabelas (Índice)	Fragmentação (REBUILD)	
	(%) Antes	(%) Depois
TbLds (Pk_TbLds)	43,56	0,00
TbLdsPar (I_LdsTtLdsPar)	70,24	0,07

TABELA 1 – Resultado da aplicação do cenário 1 (Desfragmentação de índices)
Fonte : Autoria própria

4.3.3 Cenário 2

No cenário 2, que tem como base o cenário 1, foram criadas três novas tabelas no banco de dados do sistema ERP contendo os dados de cada uma das três tabelas das bancadas de calibração. Estas novas tabelas serão “populadas” por gatilhos (TRIGGERS) após a realização de uma inserção em suas tabelas originais pelo software de calibração, esta técnica utilizada é denominada de refatoração.

Essa técnica de refatoração, usando gatilhos, irá proporcionar um ganho considerável de desempenho no que diz respeito à leitura e perda no que se diz respeito à escrita, mas no ambiente, escrita não é um gargalo no momento.

Nestas novas tabelas também foram criados índices, mais uma das técnicas apresentadas neste trabalho, para o campo: número de série objetivando melhoria no tempo de pesquisa aos dados, a escolha da indexação por este campo específico se deu devido às integrações serem realizadas por este e também pelas consultas utilizarem com frequência este campo como parâmetro. Sempre é bom comentar que o uso excessivo de índices podem trazer resultados opostos aos desejados.

Cenários comparados	Média de tempo de pesquisa (base de 14 consultas)
Normal (Cenário 1)	00h:02m:04s
Aplicando a Refatoração e Índices (Cenário 2)	00h:00m:24s

TABELA 2 – Resultado do cenário 2 (Refatoração usando gatilho e criação de índices)

Fonte : Autoria própria

4.3.4 Cenário 3

No cenário 3, que tem como base o cenário 2, implementou-se o particionamento de tabelas utilizando o MSSQL, segundo a Microsoft (4)), diz que este recurso: permite a difusão de dados de tabela e índice por vários grupos de arquivos em partições. Uma função de partição define como as linhas de uma tabela ou índice são mapeadas para um conjunto de partições, com base nos valores de certas colunas, chamadas de colunas de particionamento. Um esquema de partição mapeia cada partição especificada pela função de partição para um grupo de arquivos. Isso permite desenvolver estratégias de arquivamento que possibilitam o dimensionamento de tabelas em grupos de arquivos e, portanto, em dispositivos físicos.

A tabela particionada foi a tabela de leituras do banco de dados do ERP (Tabela: TbLds), utilizando a função e o esquema de particionamento, pelo campo de data, conforme descrito abaixo:

Função de Particionamento utilizada: CREATE PARTITION FUNCTION F_Particao(DATETIME) AS RANGE LEFT FOR VALUES ('20061231', '20071231', '20081231', '20091231', '20101231')

Esquema de Particionamento utilizado: CREATE PARTITION SCHEME E_Particao AS PARTITION F_Particao TO ([PA2007], [P20072008], [P20082009], [P20092010], [P20102011], [PD2011])

Período	Linhas a percorrer		% do número de linhas a percorrer (usando Particionamento)	Número de partições a percorrer (usando Particionamento)
	<i>Particionamento</i>			
	Não	Sim		
2006	5.045.473	27.638	00,55	1
2007	5.045.473	20.349	00,40	1
2008	5.045.473	142.247	02,82	1
2009	5.045.473	3.095.200	61,35	1
2010	5.045.473	1.760.039	34,88	1
2006 - 2007	5.045.473	47.987	00,95	2
2007 - 2008	5.045.473	162.596	03,22	2
2008 - 2009	5.045.473	3.237.447	64,16	2
2009 - 2010	5.045.473	4.855.239	96,23	2
2008 - 2010	5.045.473	4.997.486	99,05	3

TABELA 3 – Resultado da aplicação do cenário 3 (Particionamento)

Fonte : Autoria própria

A seguir, demonstram-se os resultados obtidos com o uso das técnicas apresentadas anteriormente.

5. ANÁLISE DOS DADOS

Todos os testes foram realizados em um computador com as seguintes configurações: processador Intel® Core™ i3-530 Processor (4M Cache, 2.93 GHz), RAM de 6GB e disco SATA de 500GB e 7200 rpm. Utilizando o SGBD: Microsoft SQL Server 2008 R2 e o Sistema Operacional Windows Server 2008 R2.

Foram executadas as quatorze consultas mais utilizada no dia a dia da empresa estudada, consultas essas que são utilizadas tanto para controle interno, quanto para

órgãos fiscalizadores. Cada uma dessas consultas foi executada em cada um dos cenários expostos no trabalho e obtiveram-se os seguintes resultados:

Observação: Antes da execução de cada uma das consultas, em cada um dos cenários, foi executada uma limpeza do cache (DBCC DROPCLEANBUFFERS) e dos planos de execução (DBCC FREEPROCCACHE), para não se levar a resultados viciados.

Consultas Executadas	Número de Linhas	Cenários (Tempo de Execução das Consultas)			
		Cenário 0	Cenário 1	Cenário 2	Cenário 3
Consulta 1	5.045.473	00:03:28s	00:02:26s	00:02:13s	00:02:45s
Consulta 2	1.231.247	00:01:28s	00:01:30s	00:00:11s	00:00:16s
Consulta 3	176.501	00:00:08s	00:00:08s	00:00:03s	00:00:03s
Consulta 4	507.637	00:00:16s	00:00:16s	00:00:06s	00:00:06s
Consulta 5	5.045.505	00:03:26s	00:03:15s	00:01:05s	00:01:15s
Consulta 6	485.162	00:02:42s	00:02:28s	00:00:21s	00:00:21s
Consulta 7	496.403	00:03:00s	00:02:46s	00:00:23s	00:00:22s
Consulta 8	58.921	00:02:20s	00:02:19s	00:00:09s	00:00:09s
Consulta 9	367	00:01:16s	00:01:12s	00:00:01s	00:00:01s <
Consulta 10	138	00:01:08s	00:03:23s	00:00:03s	00:00:01s <
Consulta 11	2.323	00:02:31s	00:02:15s	00:00:08s	00:00:08s
Consulta 12	138	00:02:10s	00:02:08s	00:00:09s	00:00:07s
Consulta 13	130	00:02:12s	00:02:11s	00:00:08s	00:00:07s
Consulta 14	93.703	00:03:59s	00:02:38s	00:00:40s	00:02:17s

TABELA 4 – Comparativo de resultados (Tempo de execução das consultas)

Fonte : Autoria própria

6. CONSIDERAÇÕES FINAIS

Diante do crescente aumento do volume de dados no mundo e da competitividade do mercado, as empresas estão exigindo ainda mais informações e de uma maneira muito mais dinâmica, onde qualquer ganho de tempo já trás uma grande diferença e torna-se um diferencial em relação aos demais concorrentes.

Nosso estudo tratou de melhoria de desempenho de consultas em banco de dados, onde aplicamos técnicas de melhores práticas do mercado. A partir dos resultados acima apresentados, pode-se observar que, dentre as técnicas utilizadas, a refatoração em conjunto com a utilização de índices apresentou os melhores resultados na maioria dos testes realizados, observamos também nos resultados, que o ganho de desempenho é ainda maior quando a consulta é mais seletiva, ou seja, quando possuem parâmetros específicos e não se consulta a tabela ou tabelas, por completo.

Os resultados apresentados podem indicar oportunidades de melhoria no ajuste de desempenho e configurações no que se diz respeito ao SGBD (adição de mais recursos de memória e processamento, assim como substituição de discos rígidos por padrões mais rápidos, como por exemplo: discos SSD - Solid-State Drive) e ao sistema operacional (Atualização para uma versão mais atual), como também o aperfeiçoamento de técnicas de melhoria de desempenho, como a necessidade de detecção e ajuste de índices fragmentados.

Embora o ajuste de desempenho do SGBD tenha sido realizado com base nas melhores práticas disponibilizadas nos manuais do fabricante, acredita-se que um ajuste ainda mais elaborado pode produzir resultados ainda melhores, como por exemplo: um ajuste mais detalhado do hardware e ou sistema operacional pode proporcionar benefícios para ambas às técnicas.

7. TRABALHOS FUTUROS

Como proposta para trabalhos futuros, pode-se ampliar este estudo de caso através da análise das mesmas técnicas utilizando discos SSD, que gera um ganho considerável em relação ao disco convencional SATA. Outra proposta de estudo seria colocarmos esse mesmo cenário em um ambiente Linux, utilizando o mesmo ambiente de hardware, e como sistema gerenciador de banco de dados, o Microsoft SQL Server 2016.

6. REFERÊNCIAS

CCM (1). **Bancos de dados - Introdução**. Disponível em: <http://br.ccm.net/contents/65-bancos-de-dados-introducao>. Acesso em: 22 nov. 2016.

CURBETE, Marcelo. **Mas, afinal, o que é Desempenho?**. Disponível em: <http://www.conhecimentoestrategico.com.br/2010/03/mas-afinal-o-que-e-desempenho.html> Acesso em: 24 mar. 2015.

DEVMEDIA. **Refactoring de Banco de Dados**. Disponível em: <http://www.devmedia.com.br/refactoring-de-banco-de-dados-sql-magazine-81/18429>. Acesso em: 24 mar. 2015.

FONSECA, J. J. S. **Metodologia da pesquisa científica**. Fortaleza: UEC, 2002. Apostila.

MICROSOFT (1). **Reorganizar e recriar índices**. Disponível em: <http://technet.microsoft.com/pt-br/library/ms189858.aspx>. Acessado em: 24 mar. 2015.

MICROSOFT (2). **Desempenho do Banco de Dados**. Disponível em: <http://msdn.microsoft.com/pt-br/library/ms190619.aspx>. Acesso em: 24 mar. 2015.

MICROSOFT (3). **Arquivo e grupos de arquivos do banco de dados**. Disponível em: <http://msdn.microsoft.com/pt-br/library/ms189563.aspx>. Acesso em: 24 mar. 2015.

MICROSOFT (4). **Usando o particionamento de tabelas e índice**. Disponível em: <http://msdn.microsoft.com/pt-br/library/ms162136.aspx>. Acesso em: 24 mar. 2015.

OLIVEIRA, Silvio Luiz de. **Tratado de metodologia científica**. São Paulo: Pioneira, 2010.

SILBERSCHATZ, A.; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 6ª ed. Rio de Janeiro: Campus - Elsevier, 2012.

VERGARA, Sylvia Constant. **Projetos e relatórios de pesquisa em administração**. São Paulo: Atlas, 2010.