

Tuning em queries em Banco de Dados

Alison K. L. Leite¹, Izabella C. A. M. S. Bosco¹, Juliana M. Massoni¹, Patricia B. Ribeiro¹

¹Curso de Tecnologia em Banco de Dados - Faculdade de Tecnologia de Bauru
(FATEC)

Rua Manoel Bento da Cruz, nº 30 Quadra 3 - Centro - 17.015-171 - Bauru, SP - Brasil

{alison.leite, juliana.massoni, izabella.bosco}@fatec.sp.gov.br,
patriciabellin@yahoo.com.br

Abstract. *Due to an increasing use of information systems, it is necessary that they made were fast and secure to meet the high demand of its users. What has generated a great interest in optimization and performance. This study aims to present a method to optimize database queries and show why we optimize SQL statements. Tool developed by the Database Engine Tuning Advisor, numerous techniques for optimizing database, which led to a great improvement in performance of the same. It is concluded that the method effectively demonstrated the proposed objectives at work.*

Resumo. *Em decorrência de uma utilização crescente de sistemas de informação, fez-se necessário que estes fossem rápidos e seguros para atender à grande demanda de seus usuários. O que acabou gerando um grande interesse em otimização e melhorarias de desempenho para os mesmos. Este estudo tem como objetivo apresentar um método, que otimize consultas em banco de dados e demonstre o porquê de otimizarmos instruções Structured Query Language (SQL). Desenvolvemos através da ferramenta Database Engine Tuning Advisor, inúmeras técnicas para otimização de banco de dados, o que ocasionou uma grande melhoria na performance dos mesmos. Conclui-se então que o método demonstrou de forma eficaz os objetivos propostos no trabalho.*

1. Introdução

No início dos sistemas de informação, os bancos de dados continham pouca informação e as consultas realizadas no banco não precisavam passar por melhorias, pois através do método padrão conseguia-se a resposta no tempo em que o usuário precisava. Todavia, com o passar do tempo ocorreu um crescimento exponencial destas informações o que tornou praticamente impossível que as consultas realizadas continuassem a serem feitas da mesma maneira. Sua utilização tornou-se muito demorada, ocasionando problemas ao usuário pela perda de tempo em conseguir resgatar as informações requisitadas. Consequentemente exigiu-se a necessidade da criação de um novo método que torna-se mais fácil encontrar com sucesso e segurança a informação num banco de dados [Prado, 2012]. Fundamentando neste avanço, os profissionais da área de Tecnologia da Informação iniciaram uma busca para criação de técnicas que permitissem a otimização

de banco de dados, considerando itens como: a arquitetura do sistema, a rede, o hardware e o banco de dados em que ele armazena as suas informações. Para alcançar uma melhor performance nos bancos de dados surgiram técnicas para tuná-los, atuando especificamente nas instruções SQL e em itens que influenciam o seu desempenho. Sendo necessário analisar o plano de execução, verificar o desempenho das instruções SQL e verificar se elas realmente precisam ser tunadas. Através dessas técnicas gerou-se o conceito de *tuning* de banco de dados, método hoje crucial para encontrar com maior agilidade as informações contidas no banco de dados [Prado, 2012].

Este método tem o objetivo de minimizar o tempo de resposta e de recuperação de dados, minimizar a concorrência de acesso aos dados, otimizar a taxa de transferência de dados e otimizar a capacidade de carga do Banco de Dados.

Este estudo apresenta na seção 2 a fundamentação teórica deste estudo, abordando o conceito da técnica de *tuning* e suas características. Na seção 3 estão descritos os materiais e os métodos utilizados neste trabalho, descrevendo o sistema operacional, abordando a linguagem de programação para o desenvolvimento da aplicação proposta e descrevendo a ferramenta utilizada nos testes, bem como os *hosts* e softwares utilizados, explicando o funcionamento de cada um deles, e os resultados obtidos. Na seção 4 são descritas as considerações finais, e ao final as referências.

2. *Tuning* de dados

Para que se entenda o conceito de *Tuning* é preciso entender o que é um Sistema Gerenciador de Banco de Dados (SGBD), este se trata de uma ferramenta que possibilita aos usuários a manipulação do banco de dados, visto que a técnica de *Tuning* busca aperfeiçoar os processos de consultas através do SGBD.

Segundo Date [2004], o Sistema Gerenciador de Banco de Dados é uma camada de *software*, que fica entre o usuário e o Banco de Dados. Desta forma, Todas as solicitações de acesso ao banco de dados, como criação de tabelas, inserção de dados, visões e recuperação de dados são tratadas e manipuladas pelo SGBD.

Na etapa de desenvolvimento do Banco de Dados, não é possível avaliar o seu funcionamento, assim, somente após o banco estar operante será possível identificar suas falhas que ainda não haviam sido consideradas e aplicar as técnicas para otimização.

Conforme Couto [2006], o intuito da otimização é mostrar de forma mais rápida a informação solicitada pelo usuário. Nesta fase de aperfeiçoamento, pode se aplicar o *Tuning*, que irá otimizar o acesso aos dados, fazendo com que o tempo de resposta fornecido pela aplicação seja cada vez menor.

O *Tuning* em Banco de Dados é a sintonia (*Tuning*) ou o ajuste que será aplicado nas linhas de comando SQL, para se obter melhores resultados, assim o SQL *Tuning* tem a função de melhorar o desempenho das aplicações que serão executadas, ressaltando a importância do diagnóstico de forma clara e precisa [Proni, 2010].

Estes ajustes consistem em aprimorar a aplicação para que seja executada com mais rapidez, diminuir significativamente o tempo de resposta das transações e consultas realizadas no banco, visando aperfeiçoar o desempenho do banco de forma geral.

Segundo Baptista [2008], “a técnica de *Tuning* requer os conhecimentos no desenvolvimento da aplicação, no SGBD, no Sistema Operacional e *Hardware*”. Estes

conhecimentos são indispensáveis para que a técnica traga os resultados de forma global e eficiente, ajustando o SGBD para melhor utilização dos seus recursos.

O conceito de *Tuning* pode ser aplicado, não apenas nos comandos, mas em todo o projeto. Desta forma, os ajustes para otimização podem ser aplicados em sua estrutura lógica, estrutura física, na alocação de memória, ou seja, em todos os setores do projeto em que seja possível realizar ajustes para se obter melhorias. [Tramontina, 2008].

2.1. Funcionamento da técnica de *Tuning* em Banco de Dados

O processo de *Tuning* de Banco de Dados em um projeto engloba todo o contexto, desde alterações nos diagramas de entidade relacionamento (ER) até mudanças no hardware, segundo Tramontina [2008]. Além dos conceitos de *Tuning*, também são utilizadas ferramentas desenvolvidas para aplicar as otimizações em um banco de dados. O *SQL Tuning Advisor* (Oracle) e *SQL Server Profiler* (Microsoft) são algumas ferramentas que podem ser citadas, entre muitas distribuídas no mercado. Ao contrário das técnicas de ajuste menos automatizadas que constam em Burgo [2007], o *tuning* automático de dados auxilia ou pode até realizar os ajustes nos códigos SQL através da sua análise da condição do sistema conforme sua necessidade. Cada ferramenta pode funcionar de maneira diferente, porém com o mesmo objetivo de ajustar as aplicações para um funcionamento mais ágil dos processos.

Segundo Ikematu [2009], o funcionamento da técnica de *Tuning* em Banco de Dados consiste em realizar o ajuste em comandos onde os benefícios excedam o custo do ajuste. Desta forma, devem ser localizados os comandos que consomem mais recursos ou aqueles que são executados com mais frequência. Existem alguns fatores que precisam ser elencados com a finalidade de identificar os pontos críticos para que possam ser otimizados, tais como a verificação das consultas que estão apresentando lentidão em seu tempo de resposta e a monitoria de todas as sessões que estão ativas, ou seja, em execução no banco de dados.

O foco para aplicar o *Tuning* deve ser sempre no ajuste de comandos em que os benefícios sejam maiores do que o custo do ajuste, visto que é inútil ajustar comandos SQL que para projetos ineficientes na aplicação [Ikematu, 2009]. Após a identificação de tais pontos define-se então a melhor maneira para se aplicar a Técnica de *Tuning* em Banco de Dados, utilizando ferramentas que aplicam as otimizações de forma automática ou de forma manual através das linhas de comando no próprio banco.

2.2. Database Engine Tuning Advisor

De acordo com Rosa [2008] o *Database Engine Tuning Advisor* é uma ferramenta que através da análise sobre as queries de entrada no analisador (*workload*), comparando o custo de cada uma para a base de dados, indica qual a melhor estrutura para suas tabelas, particionamentos ou *views* indexadas para resolução de problemas de desempenho. Segundo Alves [2013], esta ferramenta é utilizada quando já há bastante carga na base e há a necessidade de se obter algum tipo de ajuda na criação de objetos para otimização.

3. Materiais e Métodos

Para o desenvolvimento do estudo foi realizada uma pesquisa bibliográfica sobre *tuning* em banco dados, quais bancos em que o conceito pode ser aplicado, suas principais técnicas, métodos e sistema operacional em que teria uma melhor performance.

Para efetuar os testes do método desenvolvido, foi utilizada uma rede local composta por um *host*, sistema operacional *Windows 8.1 Pro* da Microsoft, caracterizado pela agilidade de execução multitarefa com os aplicativos. Este sistema operacional de 64 bits possui aplicativos exclusivos para utilização nesta plataforma sendo necessária uma conta na Microsoft para a utilização de alguns recursos. Sua compatibilidade com a ferramenta utilizada para a realização dos testes foi fundamental para obtenção dos resultados, pois a mesma foi desenvolvida pela Microsoft, não existindo disponibilização de versões da mesma para o sistema operacional Linux. Através da ferramenta *Database Engine Tuning Advisor* é possível filtrar suas análises estreitando-as a analisar somente consultas que o usuário necessite, assim não serão criados nem poucos, nem muitos índices, somente o necessário, contribuindo para que o desenvolvedor do banco não tenha o trabalho de analisar todas as consultas da base de dados sozinho. “Índices são estruturas que possuem algoritmos otimizados para acessar dados. Podemos dizer que um índice é uma cópia da tabela original, mas armazenada de forma inteligente.” [Rosa, 2008].

O banco de dados escolhido foi o *Microsoft SQL Server* em que se usa a linguagem de programação SQL (*Structured Query Language*) padrão para bases de dados relacionais. Esta linguagem de consulta estruturada realiza o gerenciamento dos dados e interage com os bancos de dados relacionais de forma declarativa, onde é preciso indicar ao SGBD o que deve ser executado. O SQL é compatível com a maioria das plataformas atuais e não exige muitos recursos do hardware para apresentar estabilidade e desempenho, por este motivo está presente de forma visível ou não nos sistemas de banco de dados.

Sua sintaxe corresponde ao conjunto de regras com base nas quais os elementos de uma linguagem são corretamente combinados e possui instruções contidas em cláusulas, tais como as que utilizamos em nossos testes dentre elas: SELECT, FROM, WHERE, ORDER BY, GROUP BY, HAVING, CASE, INNER JOIN entre outras.

O desempenho dos processos foram obtidos através da análise da comparação entre o tempo de execução da consulta SQL sem a utilização das técnicas de *tuning* e com a utilização das mesmas, e como base foi utilizado banco de dados DADOSMP_TST, retirado de empresa que utiliza o ERP TOTVS *Protheus*, que possui inúmeras tabelas com vários registros, o que possibilitou uma melhor análise das consultas SQL e da aplicação das técnicas de *tuning*.

3.1 Resultados

As consultas que serão apresentadas a seguir são utilizadas para demonstrar o uso do método de *tuning* de dados em um banco *Microsoft SQL Server*.

A Figura 1 demonstra a criação de uma consulta SQL sem a utilização da técnica de *tuning*, com o uso do comando INNER JOIN para combinação dos registros entre as tabelas, SUM() para somatória de valores, COUNT() para contagem de registros. O tempo para execução da consulta foi de 00:02:25, destacado na imagem.

alta qualidade; em alguns casos, é necessário criar estatísticas adicionais ou modificar o design da consulta para obter melhores resultados.

```

CONSULTA_001 COM...MP_TST (sa (55))
use [DADOSMP_TST]
go

SET ANSI_PADDING ON
go

CREATE NONCLUSTERED INDEX [_dta_index_SE5010_11_1996742366__K81_K13_K82_K21_K23_K3_K24_K25_K16_K5] ON [dbo].[SE5010]
(
    [D_E_L_E_T_] ASC,
    [E5_RECPEG] ASC,
    [R_E_C_N_O_] ASC,
    [E5_NUMERO] ASC,
    [E5_PREFIXO] ASC,
    [E5_PARCELA] ASC,
    [E5_TIPO] ASC,
    [E5_CLIFOR] ASC,
    [E5_LOJA] ASC,
    [E5_TIPODOC] ASC,
    [E5_VALOR] ASC
)WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]
go

CREATE STATISTICS [_dta_stat_1996742366_21_81] ON [dbo].[SE5010]([E5_PREFIXO], [D_E_L_E_T_])
go

CREATE STATISTICS [_dta_stat_1996742366_5_81] ON [dbo].[SE5010]([E5_VALOR], [D_E_L_E_T_])
go

CREATE STATISTICS [_dta_stat_1996742366_16_81] ON [dbo].[SE5010]([E5_TIPODOC], [D_E_L_E_T_])
go

CREATE STATISTICS [_dta_stat_1996742366_82_81_5] ON [dbo].[SE5010]([R_E_C_N_O_], [D_E_L_E_T_], [E5_VALOR])
go
    
```

Figura 3. Criação de índices e *statistics* recomendados pela Ferramenta
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

A Figura 4 demonstra a execução da consulta após a aplicação das recomendações da ferramenta realizadas na Figura 3. O tempo de execução foi de 00:00:34. Nota-se a diminuição de 00:01:51 no tempo em relação a consulta realizada na Figura 1 sem a utilização do *tuning* de dados.

```

SELECT
    SA1.A1_NOME RAZAO_SOCIAL,
    SF2.F2_SERIE SERIE_NOTA,
    SF2.F2_DOC NUMERO_NOTA,
    SUM(SD2.D2_TOTAL) VALOR_TOTAL,
    COUNT(SD2.D2_DOC) QTDE_ITENS,
    CASE SF4.F4_DUPLOC WHEN 'S' THEN 'SIM' ELSE 'NAO' END POSSUI_DUPLICATA_FINANCEIRA,
    COUNT(SE1.E1_NUM) QTDE_PARCELAS_DUPLICATA,
    ISNULL(SUM(CASE SE5.E5_TIPODOC WHEN 'ES' THEN (SE5.E5_VALOR * -1) ELSE SE5.E5_VALOR END),0) VALOR_PAGO,
    ISNULL(SE1.E1_SALDO,0) SALDO_DUPLICATA
FROM
    SF2010 SF2
INNER JOIN SA1010 SA1 ON
    SA1.A1_COD = SF2.F2_CLIENTE AND
    SA1.A1_LOJA = SF2.F2_LOJA AND
    SA1.D_E_L_E_T_ = ''
INNER JOIN SD2010 SD2 ON
    
```

	RAZAO_SOCIAL	SERIE_NOTA	NUMERO_NOTA	VALOR_TOTAL	QTDE_ITENS	POSSUI_DUPLICATA_FINANCEIRA	QTDE_PARCELAS_DUPLICATA
1	AILTON MANOEL MARTINS	CLC	109740	27	1	SIM	1
2	AILTON MANOEL MARTINS	CLC	110300	27	1	SIM	1
3	CEOB CENTRO ODONTOLOGICO DE BAURU	CLA	002791	18	1	SIM	1
4	CEOB CENTRO ODONTOLOGICO DE BAURU	CLC	089119	18	1	SIM	1
5	EMBRASIL EMPRESA BRASILEIRA DISTRIBUIDORA	CLA	001907	292,5	1	SIM	1
6	G & R NUCLEO DE FORMACAO DE ATLETAS LTDA	UNI	012654	5085,6	1	SIM	1
7	G & R NUCLEO DE FORMACAO DE ATLETAS LTDA	UNI	012654	35599,2	7	SIM	7
8	JAQUELINE GONCALVES MATOS	CLC	121094	13,5	1	SIM	1
9	JAQUELINE GONCALVES MATOS	CLC	121796	10,5	1	SIM	1
10	LAUDELINA ROSA	CLC	000004	12,0	1	SIM	1

100 %

LOCALHOST (12.0 RTM) | sa (54) | DADOSMP_TST | 00:00:34 | 133762 rows

Query executed successfully.

Figura 4. Consulta 1 com utilização da técnica de *tuning* de dados
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

A Figura 5, demonstra uma nova consulta *sql* ao banco de dados sem a utilização das recomendações da ferramenta, com o uso dos comandos: INNER JOIN para combinação dos registros entre as tabelas, WHERE para filtrar a consulta, GROUP BY para que o agrupamento dos registros seja feito pelos campos especificados e ORDER BY comando que ordena os registros na consulta. O tempo para execução da consulta foi de 00:00:34, destacado na imagem.

```

SELECT
  A1_NOME RAZAO_SOCIAL,
  A1_NREDUZ NOME_FANTASIA,
  COUNT(E1_NUM) QTDE_COMPRAS,
  'R$ ' + CAST(MAX(E1_VALOR) AS VARCHAR(10)) MAIOR_COMPRA
FROM
  SA1010 SA1 --CLIENTES
  INNER JOIN SE1010 SE1 ON --CONTAS A RECEBER
    E1_CLIENTE = A1_COD AND
    E1_LOJA = A1_LOJA AND
    SE1_D_E_L_E_T_ = ' '
WHERE
  A1_MSBLQL <> '1' AND --SOMENTE CLIENTE NÃO BLOQUEADOS
  E1_SALDO = 0 AND --SOMENTE PAGOS
  SA1_D_E_L_E_T_ = ' '
GROUP BY
  A1_NOME,
  A1_NREDUZ
ORDER BY
  A1_NOME
  
```

	RAZAO_SOCIAL	NOME_FANTASIA	QTDE_COMPRAS	MAIOR_COMPRA
1	AILTON MANOEL MARTINS	AILTON MANOEL MARTINS	2	R\$ 27
2	CEOB CENTRO ODONTOLOGICO DE BAURU	CEOB CENTRO ODONTOLOGICO DE BAURU	2	R\$ 18
3	EMBRASIL EMPRESA BRASILEIRA DISTRIBUIDORA	EMBRASIL EMPRESA BRASILEIRA DISTRIBUIDORA	1	R\$ 292.5
4	G & R NUCLEO DE FORMACAO DE ATLETAS LTDA	NAFA.7 NUCLEO AVANÇADO DE FORMACAO DE ATLETAS	8	R\$ 636
5	IAQUINI INEFGONCALVES MATOS			

Query executed successfully. LOCALHOST (12.0 RTM) sa (54) DADOSMP_TST 00:00:34 21015 rows

Figura 5. Consulta 2 com aplicação de boas práticas
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

Na Figura 6, são exibidas as recomendações da ferramenta *Database Engine Tuning Advisor* para otimizar o tempo de execução e o processamento da consulta, visualizada na Figura 5. As recomendações são: criação de índices e estatísticas, conforme campos utilizados na consulta 2.

Detalhes	Esquema de Partição	Tamanho (KB)	Definição
			(#A1_LOJA ID E L E T)
			(#A1_COD A1_LOJA)
			(#A1_NOME ID E L E T)
			(#A1_LOJA IR E C N O I D E L E T)
			(ID E L E T I IR E C N O I A1_COD)
K4		4320	(ID E L E T I asc A1_LOJA asc IR E C N O I asc A1_COD asc A1_NOME asc)
			(IR E C N O I A1_COD A1_LOJA ID E L E T I A1_NOME)
			(ID E L E T I A1_COD A1_LOJA)
			(ID2_LOJA ID E L E T I ID2_TES)
			(ID2_CLIENTE ID2_LOJA ID2_FILIAL ID2_SERIE ID2_DOCI ID E L E T I)
			(ID2_SERIE ID E L E T I ID2_TES ID2_CLIENTE ID2_LOJA)
			(ID2_DOCI ID E L E T I ID2_TES ID2_CLIENTE ID2_LOJA ID2_FILIAL)
			(ID2_CLIENTE ID2_LOJA ID E L E T I)
			(ID E L E T I asc ID2_CLIENTE asc ID2_LOJA asc ID2_FILIAL asc ID2_SERIE asc ID2_DOCI asc ID2_TES asc include (ID2_TOTAL)
			(ID2_CLIENTE ID E L E T I ID2_TES ID2_LOJA ID2_FILIAL ID2_SERIE ID2_DOCI)
			(ID2_TES ID2_CLIENTE ID2_LOJA ID2_FILIAL ID2_SERIE ID2_DOCI)
			(#E1_PREFIXO IE1_NUMI IE1_CLIENTE IE1_LOJA IE1_PARCELA IE1_TIPO)
			(#E1_NUMI ID E L E T I)
			(#E1_LOJA ID E L E T I IE1_PREFIXO IE1_NUMI)
			(#E1_PREFIXO IE1_NUMI IE1_PARCELA IE1_TIPO IE1_CLIENTE)
6 K7 K8 K33		18248	(ID E L E T I asc IE1_PREFIXO asc IE1_NUMI asc IE1_PARCELA asc IE1_TIPO asc IE1_CLIENTE asc IE1_LOJA asc IE1_SALDO asc)

Figura 6. Consulta 2 com recomendações da ferramenta
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

Na Figura 7, descreve a criação dos comandos recomendados pela ferramenta na Figura 6, os índices e as *statistics*.

```

use [DADOSMP_TST]
go

SET ANSI_PADDING ON
go

CREATE NONCLUSTERED INDEX [_dta_index_SE1010_7_410874418_K33_K226_7_8_13] ON [dbo].[SE1010]
(
    [E1_SALDO] ASC,
    [D_E_L_E_T_] ASC
)
INCLUDE (
    [E1_CLIENTE],
    [E1_LOJA],
    [E1_VALOR]) WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]
go

CREATE STATISTICS [_dta_stat_410874418_7_8_33] ON [dbo].[SE1010]([E1_CLIENTE], [E1_LOJA], [E1_SALDO])
go

CREATE STATISTICS [_dta_stat_410874418_226_7_8] ON [dbo].[SE1010]([D_E_L_E_T_], [E1_CLIENTE], [E1_LOJA])
go

CREATE STATISTICS [_dta_stat_410874418_33_226_7_8] ON [dbo].[SE1010]([E1_SALDO], [D_E_L_E_T_], [E1_CLIENTE], [E1_LOJA])
go

SET ANSI_PADDING ON
go

CREATE NONCLUSTERED INDEX [_dta_index_SA1010_7_887843121_K181_K158_K2_K3_K4_K6] ON [dbo].[SA1010]
(
    [D_E_L_E_T_] ASC,

```

Figura 7. Criação de índices e *statistics* recomendados pela Ferramenta
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

Na Figura 8, demonstra a execução da consulta após a aplicação das recomendações da ferramenta realizadas na Figura 7. O tempo de execução foi de 00:00:13. Nota-se a diminuição de 00:00:11 no tempo em relação a consulta realizada na Figura 5 sem a utilização do *tuning* de dados.

The screenshot shows a SQL query being executed in SQL Server Enterprise Manager. The query is a SELECT statement with a complex join and filter. The results are displayed in a table with 8 rows. The performance metrics at the bottom indicate the query was executed successfully on LOCALHOST (12.0 RTM) in 00:00:13, returning 21015 rows.

RAZAO_SOCIAL	NOME_FANTASIA	QTDE_COMPRAS	MAIOR_COMPRA
AILTON MANOEL MARTINS	AILTON MANOEL MARTINS	2	R\$ 27
CEOB CENTRO ODONTOLÓGICO DE BAURU	CEOB CENTRO ODONTOLÓGICO DE BAURU	2	R\$ 18
EMBRASIL EMPRESA BRASILEIRA DISTRIBUIDORA	EMBRASIL EMPRESA BRASILEIRA DISTRIBUIDORA	1	R\$ 292.5
G & R NUCLEO DE FORMACAO DE ATLETAS LTDA	NAFA 7 NUCLEO AVANÇADO DE FORMACAO DE ATLETAS	8	R\$ 636
JAQUELINE GONCALVES MATOS	JAQUELINE GONCALVES MATOS	1	R\$ 13.5
LAUDELINA ROSA	LAUDELINA ROSA	3	R\$ 13.5
MARTINS DOMINGOS DE JESUS SANTOS PANTAI FON	MARTINS DOMINGOS DE JESUS SANTOS PANTAI FON	1	R\$ 62.4
SANDRA MARA VICENTE & CIA LTDA- ME			

Query executed successfully. LOCALHOST (12.0 RTM) sa (54) DADOSMP_TST 00:00:13 21015 rows

Figura 8. Consulta 2 com aplicação do *tuning*
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

A Figura 9, demonstra uma terceira consulta *SQL* ao banco de dados sem a utilização das recomendações da ferramenta, com o uso dos comandos: *INNER JOIN* para combinação dos registros entre as tabelas, *WHERE* para filtrar a consulta, *GROUP BY* para que o agrupamento dos registros seja feito pelos campos especificados e *ORDER BY* comando que ordena os registros na consulta. O tempo para execução da consulta foi de 00:00:35, destacado na imagem.

```

SELECT
  B1_DESC PRODUTO,
  A2_NOME FORNECEDOR,
  COUNT(D1_COD) VEZES_COMPRADO,
  SUM(D1_QUANT) QTDE_COMPRADA,
  'R$ ' + CAST(MAX(D1_VUNIT) AS VARCHAR(10)) MAIOR_VALOR_PAGO_POR_UNIDADE
FROM
  SB1020 SB1 --PRODUTOS
  INNER JOIN SD1020 SD1 ON --DOCUMENTOS DE ENTRADA
    D1_COD = B1_COD AND
    SD1_D_E_L_E_T_ = ' '
  INNER JOIN SA2020 SA2 ON --FORNECEDORES
    A2_COD = D1_FORNECE AND
    A2_LOJA = D1_LOJA AND
    SA2_D_E_L_E_T_ = ' '
WHERE
  B1_MSBLQL <> '1' AND --SOMENTE NÃO BLOQUEADOS
  A2_MSBLQL <> '1' AND --SOMENTE NÃO BLOQUEADOS
  SB1_D_E_L_E_T_ = ' '
GROUP BY
  B1_DESC,
  A2_NOME
ORDER BY
  B1_DESC,
  A2_NOME
  
```

PRODUTO	FORNECEDOR	VEZES_COMPRADO	QTDE_COMPRADA	MAIOR_VALOR_PAGO_POR_UNIDADE
1/2 BLOCO 12X20X20CM	CERAMICA MEDEANA LTDA	8	8,1	R\$ 600
ABRACADEIRA	AGROSOLO BAURU AGROPECUARIA LTDA	1	8	R\$ 1.95
ABRACADEIRA	ELDOR/			
ABRACADEIRA	IARILE			

Query executed successfully. LOCALHOST (12.0 RTM) | sa (53) | DADOSMP_TST | 00:00:35 | 6480 rows

Figura 9. Consulta 3 com aplicação de boas práticas
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

Na Figura 10, são exibidas as recomendações da ferramenta *Database Engine Tuning Advisor* para otimizar o tempo de execução e o processamento da consulta 3 visualizada na Figura 9. As recomendações são: criação de índices e estatísticas, conforme campos utilizados na consulta 3, Figura 9.

Detalhes	Esquema de Partição	Tamanho (KB)	Definição
			(SA1 LOJA) (D E L E T I)
			(SA1 COD) (SA1 LOJA)
			(SA1 NOME) (D E L E T I)
			(SA1 LOJA) (R E C N O I) (D E L E T I)
			(D E L E T I) (R E C N O I) (SA1 COD)
			(D E L E T I) asc, (SA1 LOJA) asc, (R E C N O I) asc, (SA1 COD) asc, (SA1 NOME) asc
			(R E C N O I) (SA1 COD) (SA1 LOJA) (D E L E T I) (SA1 NOME)
			(D E L E T I) (SA1 COD) (SA1 LOJA)
			(D2 LOJA) (D E L E T I) (D2 TESI)
			(D2 CLIENTE) (D2 LOJA) (D2 FILIAL) (D2 SERIE) (D2 DOC) (D E L E T I)
			(D2 SERIE) (D E L E T I) (D2 TESI) (D2 CLIENTE) (D2 LOJA)
			(D2 DOC) (D E L E T I) (D2 TESI) (D2 CLIENTE) (D2 LOJA) (D2 FILIAL)
			(D2 CLIENTE) (D2 LOJA) (D E L E T I)
			(D2 TESI) (D E L E T I)
			(D E L E T I) asc, (D2 CLIENTE) asc, (D2 LOJA) asc, (D2 FILIAL) asc, (D2 SERIE) asc, (D2 DOC) asc, (D2 TESI) asc include (D2 TOTAL)
			(D2 CLIENTE) (D E L E T I) (D2 TESI) (D2 LOJA) (D2 FILIAL) (D2 SERIE) (D2 DOC)
			(D2 TESI) (D2 CLIENTE) (D2 LOJA) (D2 FILIAL) (D2 SERIE) (D2 DOC)
			(E1 PREFIXO) (E1 NUM) (E1 CLIENTE) (E1 LOJA) (E1 PARCELA) (E1 TIPO)
			(E1 NUM) (D E L E T I)
			(E1 LOJA) (D E L E T I) (E1 PREFIXO) (E1 NUM)
			(E1 PREFIXO) (E1 NUM) (E1 PARCELA) (E1 TIPO) (E1 CLIENTE)
			(D E L E T I) asc, (E1 PREFIXO) asc, (E1 NUM) asc, (E1 PARCELA) asc, (E1 TIPO) asc, (E1 CLIENTE) asc, (E1 LOJA) asc, (E1 SALDO) asc

Figura 10. Consulta 3 com recomendações da ferramenta
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

A Figura 11, exibe a aplicação das recomendações feitas pela ferramenta para a otimização do tempo para a criação da consulta 3 na Figura 10, através da criação de índices e estatísticas.

```

CONSULTA_001 COM...MP_TST (sa (55))
use [DADOSMP_TST]
go

SET ANSI_PADDING ON
go

CREATE NONCLUSTERED INDEX [_dta_index_SE5010_11_1996742366__K81_K13_K82_K22_K21_K23_K3_K24_K25_K16_K5] ON [dbo].[SE5010]
(
    [D_E_L_E_T_] ASC,
    [E5_RECPCAG] ASC,
    [R_E_C_N_O_] ASC,
    [E5_NUMERO] ASC,
    [E5_PREFIXO] ASC,
    [E5_PARCELA] ASC,
    [E5_TIPO] ASC,
    [E5_CLIFOR] ASC,
    [E5_LOJA] ASC,
    [E5_TIPODOC] ASC,
    [E5_VALOR] ASC
) WITH (SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]
go

CREATE STATISTICS [_dta_stat_1996742366_21_81] ON [dbo].[SE5010]([E5_PREFIXO], [D_E_L_E_T_])
go

CREATE STATISTICS [_dta_stat_1996742366_5_81] ON [dbo].[SE5010]([E5_VALOR], [D_E_L_E_T_])
go

CREATE STATISTICS [_dta_stat_1996742366_16_81] ON [dbo].[SE5010]([E5_TIPODOC], [D_E_L_E_T_])
go

CREATE STATISTICS [_dta_stat_1996742366_82_81_5] ON [dbo].[SE5010]([R_E_C_N_O_], [D_E_L_E_T_], [E5_VALOR])
go
    
```

Figura 11. Criação de índices recomendados pela ferramenta
Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

Na Figura 12, demonstra a execução da consulta 3, Figura 10, após a aplicação das recomendações da ferramenta realizadas na Figura 11. O tempo de execução foi de 00:00:08. Nota-se a diminuição de 00:00:27 no tempo em relação a consulta realizada na Figura 9 sem a utilização do *tuning* de dados.

```

SELECT
    B1_DESC PRODUTO,
    A2_NOME FORNECEDOR,
    COUNT(D1_COD) VEZES_COMPRADO,
    SUM(D1_QUANT) QTDE_COMPRADA,
    'R$ ' + CAST(MAX(D1_VUNIT) AS VARCHAR(10)) MAIOR_VALOR_PAGO_POR_UNIDADE
FROM
    SB1020 SB1 --PRODUTOS
INNER JOIN SD1020 SD1 ON --DOCUMENTOS DE ENTRADA
    D1_COD = B1_COD AND
    SD1_D_E_L_E_T_ = ' '
INNER JOIN SA2020 SA2 ON --FORNECEDORES
    A2_COD = D1_FORNECE AND
    A2_LOJA = D1_LOJA AND
    SA2_D_E_L_E_T_ = ' '
WHERE
    B1_MSBLQL <> '1' AND --SOMENTE NÃO BLOQUEADOS
    A2_MSBLQL <> '1' AND --SOMENTE NÃO BLOQUEADOS
    SB1_D_E_L_E_T_ = ' '
GROUP BY
    B1_DESC,
    A2_NOME
ORDER BY
    B1_DESC,
    A2_NOME
    
```

PRODUTO	FORNECEDOR	VEZES_COMPRADO	QTDE_COMPRADA	MAIOR_VALOR_PAGO_POR_UNIDADE
1/2 BLOCO 12X20X20CM	CERAMICA MEDEANA LTDA	8	8,1	R\$ 600
ABRACADEIRA	AGROSOLO BAURU AGROPECUARIA LTDA	1	8	R\$ 1.95
ABRACADEIRA	ELDOF			

Query executed successfully. LOCALHOST (12.0 RTM) | sa (54) | DADOSMP_TST | 00:00:08 | 6480 rows

Figura 12. Consulta 3 com aplicação do *tuning* de dados

Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

A Tabela 1 apresenta as comparações dos resultados obtidos nas consultas realizadas através da ferramenta *Database Engine Tuning Advisor*. Demonstrando que houve ganho de desempenho e performance ao aplicarmos as técnicas de *tuning* e otimização do tempo na execução de cada consulta.

Tabela 1: Comparação dos resultados.

Consulta	Tempo Sem Tuning	Tempo Após Tuning	Diferença de Tempo	Quantidade de Registros
001	00:02:25	00:00:34	00:01:51	133.762
002	00:00:34	00:00:13	00:00:11	21.015
003	00:00:35	00:00:08	00:00:27	6.480

Fonte: Alison Leite, Izabella Bosco e Juliana Massoni

4. Conclusão

O método desenvolvido demonstrou de forma eficiente e eficaz os objetivos propostos no trabalho. Os testes realizados através das técnicas com a ferramenta *Database Engine Tuning Advisor* oferecem ao usuário da área de tecnologia da informação, seja um desenvolvedor ou administrador de banco de dados, um conjunto de informações suficientes para mostrar através das comparações com os resultados obtidos constatar a importância da aplicação do método de *tuning* em banco de dados.

O mercado atualmente disponibiliza inúmeras ferramentas que oferecem a possibilidade dos usuários desenvolverem o método de *tuning* em banco de dados. É certo como tendência que os profissionais da área façam cada vez mais o uso do método para facilitar o desempenho e a estabilidade de seus bancos de dados.

Como trabalho futuro será possível o desenvolvimento de uma nova ferramenta em que além de trazer “dicas” para realização *tuning*, possa automaticamente gerar uma nova consulta que crie índices e realize boas práticas como o uso dos comandos *case*, *where* e etc.

5. Referências

- Alves, G. A. (2013). Artigo: “Melhore o desempenho de seu banco com o Tuning Advisor-SQL Magazine 78”. Rio de Janeiro. <<http://www.devmedia.com.br/melhore-o-desempenho-de-seu-banco-com-o-tuning-advisor-sql-magazine-78/17723#ixzz3HDeTmbg1>>.
- Baptista, C. de S. (2008) Administração de Sistemas de Gestão de Banco de Dados. <www.dsc.ufcg.edu.br/~baptista/cursos/ABD/ADM1.ppt>.
- Burgo, A. F. (2007). Tuning de Banco de Dados em um software de CRM, <<http://www2.dc.uel.br/nourau/document/?view=467>>.
- Coradini, T.P; Cantarelli, G. S. (2013). Avaliação de Desempenho de Ferramentas para Tuning em Banco de Dados. Santa Maria. <<http://sites.unifra.br/Portals/36/Tecnologicas%202012-2/08.pdf>>.

- Couto, E. (2006). Artigo: Aumentando a Performance da Aplicação Através da Otimização de SQL. iMasters.<http://imasters.com.br/artigo/4055/bancodedados/aumentando_a_performance_da_aplicacao_atraves_da_otimizacao_de_sql/>.
- Date, C. J. (2004). Introdução a Sistemas de Bancos de Dados. Campus, Rio de Janeiro.
- Ikematu, R. S. (2009). Artigo: “Realizando tuning na base de aplicações”. Celepar. <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1592>>.
- Prado, H. N. D. (2012). Artigo: “Tuning de SQL em Banco de dados Oracle” SQL Magazine.
- Proni, R. P. (2013). Artigo: SQL Magazine: Oracle Benchmark: Linux x Windows. <<http://www.devmedia.com.br/oracle-benchmark-linux-x-windows-artigo-revista-sql-magazine-87/21045>>.
- Rosa, L. M. (2008). Artigo: “Tuning Index com o DTA”. Porto Alegre. <<http://www.linhadecodigo.com.br/artigo/3290/tunning-index-com-o-dta.aspx#ixzz3HDgUBgJg>>.
- Santos, J. C; SILVA, A.P.S. (2013). Otimização e Performance de Banco de Dados Utilizando SQL Tuning. Paranaíba. <<http://web.unipar.br/seinpar/2013/artigos/Jessica%20Correa%20dos%20Santos.pdf>>.
- Tramontina, G. B. (2008). Database Tuning: Configurando o Interbase e o PostgresSQL. Campinas. <<http://www.ic.unicamp.br/~geovane/mo410-091/Ch20-ConfigInterbasePosgres-art.pdf>>.