

PS-503

DATABASE MANAGEMENT SYSTEMS (DBMS) AND HORIZONTAL PARTITIONING IN TABLES: PERFORMANCE ANALYSIS IN A STEEL COMPANY

Priscila de Jesus Papazissis Matuck (Universidade FUMEC, MG, Brasil) – priscila.papazissis@fumec.br

Marcos Vinícios da Silva (Universidade FUMEC, MG, Brasil) – vinicios.m.silva@gmail.com

Daniel Jardim Pardini (Universidade FUMEC, MG, Brasil) – pardini@fumec.br

This work analyses the performance of a production database in a steel company, before and after the implementation of a partitioning technique named horizontal partitioning. This technique was applied in a system named Manufacturing Execution Systems (MES). These information and communication systems are responsible for maintain the data of the production processes in an organization. Considering the evolutionary and permanent feature of the organizational data, as the time pass, the database management systems that maintain e control these data suffer a great descent of performance, affecting directly the customers' response time. From the analysis of the obtained results, it was possible to realize that the horizontal partitioning is a process, that, in fact, contributes to improve the performance of high volume tables, despite the increase of the size that the partitioned tables will have in the hard disks.

Keywords: Data. Information. Database. Horizontal Partitioning. Otimization.

1. Introdução

O cenário competitivo moderno transformou o ambiente corporativo, fazendo com que as organizações encontrassem novos meios de melhorar o desempenho dos sistemas de informação, tornando-os confiáveis, ágeis e seguros. Para as corporações da atualidade, um fator relevante de vantagem competitiva tem sido o meio pelo qual tais empresas conseguem armazenar e reutilizar as informações que circulam, sejam elas de processos, funcionários, matérias-primas, ou qualquer outro componente organizacional. Os avanços tecnológicos que aconteceram nas últimas décadas, principalmente aqueles relacionados à Tecnologia da Informação (TI), permitiram às empresas armazenarem grandes quantidades de dados sobre seus processos. Se há trinta anos o custo de armazenar qualquer informação era alto e a capacidade para armazená-las era baixa, atualmente esse panorama se inverteu (RABELO, 2007).

Com isso, percebe-se que a maioria das empresas que buscam garantir um controle de seu negócio de forma confiável, tende a implementar algum sistema de banco de dados. Um banco de dados é composto por um conjunto de tabelas que possuem dados e objetos definidos, tais como índices, procedimentos, funções e gatilhos capazes de suportar a execução das atividades realizadas com os dados. Tais dados estão geralmente relacionados a um determinado assunto ou processo: informações de estoque para depósito de produção, funcionários, clientes, fornecedores, dentre outros (SILBERSCHATZ *et. al.*, 1999).

O meio de organização desses elementos constitui-se em um sistema de gerenciamento do banco de dados (SGBD), que é formado por um conjunto de programas que permite armazenar, excluir, modificar ou atualizar informações de determinado negócio. São diversos os fabricantes e tecnologias distintas utilizadas nessas ferramentas, que tem se tornado cada vez mais sofisticadas devido à concorrência e evolução da tecnologia. A maior parte dos sistemas de informação existentes nas empresas possui um banco de dados que é capaz de garantir a persistência das informações (OLIVEIRA, 2008).

Conforme Cintra e Capello (2005), se considerarmos a relevância estratégica que o conhecimento tem para qualquer organização, a tendência natural dos sistemas que armazenam as informações das empresas têm sido o crescimento exponencial de suas bases dados. Dessa forma, as corporações têm enfrentado um problema de ordem tecnológica, considerando o volume de dados gerado por cada sistema implementado. Quanto maior a base de dados, maior a tendência à diminuição do tempo de resposta ao acesso aos dados. Considerando a ampla utilização dos sistemas de informação, e a dificuldade em gerenciar grandes volumes de dados sem perder o tempo de resposta aceitável ao utilizá-los, surge então a necessidade de métodos capazes de manter o tempo de resposta adequado frente ao intenso volume de dados gerados pelas transações empresariais.

Dessa forma, nessa pesquisa será explorada uma das técnicas que possibilitam uma possível melhoria do tempo de resposta ao acesso a grandes volumes de dados, denominada particionamento horizontal em tabelas. Essa técnica, ainda pouco explorada em estudos de tecnologia da informação (STONEBRAKER e CATTELL, 2008), pode auxiliar as organizações a conseguir variações no desempenho de acesso aos dados corporativos. O objetivo é analisar o desempenho do sistema em resposta à implementação do particionamento horizontal em tabelas, considerando um SGBD instalado em uma empresa siderúrgica mineira.

2. Problema de Pesquisa

Nesse estudo é proposta uma análise que possibilitará medir o aumento ou a diminuição do desempenho de um SGBD, a partir da aplicação do particionamento horizontal em um banco de dados organizacional. Considerando a necessidade que as organizações têm em melhorar o tempo de acesso aos dados corporativos, objetiva-se, ainda, levantar as considerações sobre essa prática, após sua aplicação. Nesse sentido, pretende-se responder ao seguinte problema de pesquisa: *quais as percepções sobre a utilização do particionamento horizontal de tabelas, como forma de melhorar o desempenho de acesso aos dados de um sistema de informação?*

Espera-se dotar essa questão de respostas que possam auxiliar as empresas a tomarem decisões a acerca da informatização de seus processos e consequente armazenamento desses dados em bancos de dados organizados. A partir da medição do tempo de resposta dos processos corporativos e consequente análise das percepções dos envolvidos, será feita a implementação das técnicas de particionamento horizontal de tabelas em um SGBD empresarial. Para melhor compreensão das razões de utilização dessas técnicas, verifica-se, no próximo tópico, a revisão da literatura.

3. Revisão da Literatura

Com o objetivo de entender melhor o mecanismo de funcionamento de um banco de dados e suas possíveis aplicações, é necessário delinear alguns conceitos.

3.1. Dado, Informação e Conhecimento

Dado é qualquer elemento identificado em sua forma bruta que, por si só, não induz à compreensão de determinado fato ou situação (OLIVEIRA, 2006). Seltzer (2008) define dado como uma representação simbólica quantificável. Assim, um texto é um dado. Mas uma foto também é um dado, pois é possível quantificá-la reduzindo-a a símbolos – é possível digitalizá-la em um *scanner* e armazená-la em um computador, imprimindo-a posteriormente de modo que praticamente não haja distinção da foto original.

Dado é qualquer indício ou registro que permite identificar alguma característica de uma entidade ou evento. Dados não são somente o resultado da intenção do registro de algo, mas podem ser obtidos a partir de um som, uma marca, a sombra ou o aspecto de algum objeto. Para ser considerado dado, não é necessário estar armazenado em algum repositório físico – dados podem ser transmitidos (SELTZER, 2008). Será definido agora, ao conceito de informação.

A origem etimológica da palavra informação vem do latim “*informatio*”, que significa “em forma”, sugerindo a “[...] ação de formar, representar, esboçar, planejar, ou mesmo conceber uma ideia” (CUNHA, 1997). Sendo assim, a informação é uma abstração informal (isto é, que não pode ser formalizada por meio de uma teoria lógica ou matemática). Informação é o dado trabalhado que permite ao indivíduo, por exemplo, tomar decisões organizacionais (OLIVEIRA, 2006). Dessa forma, a informação consiste no dado configurado de forma adequada ao entendimento e a reutilização pelo ser humano (ROSINI e PALMISANO, 2003). As informações são o resultado dos dados devidamente tratados, classificados e relacionáveis.

O conhecimento pode ser compreendido como a “capacidade de interpretar e operar sobre um conjunto de informações” (CASTELLS, 1999). Tal capacidade é criada a partir das relações tecidas sobre esse conjunto de dados que, combinados entre si (incluindo experiências, impressões, valores, crenças, etc.), possibilitam compreender o conhecimento e produzir inferências a partir dele. Conforme Silva (2004), esses conceitos se organizam de forma hierárquica, onde os dados, uma vez “combinados numa estrutura compreensível” se tornam informação, que por seguindo a mesma lógica, podem se tornar conhecimento.

Portanto, as três definições (dado, informação e conhecimento) mantêm uma relação de interdependência, sendo uma pré-requisito da outra. Angeloni (2003), porém, explica que a ordem inversa da também é possível. Acontece quando uma informação é criada a partir do conhecimento e os dados extraídos daquela informação representam fatos isolados que por si mesmos não apresentam conteúdo compreensível. O conhecimento também não depende apenas de uma interpretação pessoal, como a informação, pois requer uma vivência do objeto do conhecimento.

A informação está associada à semântica. O conhecimento está associado com pragmática, isto é, relaciona-se com alguma coisa existente no mundo real onde se tem uma experiência direta. Seltzer (2008) também argumenta que “[...] dados são puramente sintáticos enquanto a informação contém uma semântica. Conhecimento é uma abstração interior (...) relacionada a algo existente no mundo real e do qual se tem experiência direta”. Informação e conhecimento dependem, em proporções diferenciadas, do agente interpretante para se constituírem. Sem o papel desempenhado pelo ser humano, as informações e o conhecimento são apenas possibilidades.

3.2. Banco de Dados

Um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico. Por exemplo: lista telefônica ou fichas de acervo de uma biblioteca podem constituir-se em bancos de dados. Um banco de dados pode ser entendido como um conjunto de dados integrados que tem por objetivo atender a uma comunidade de usuários. Aplicações dessa natureza são necessárias para armazenar os dados dos processos informatizados que as organizações têm implementado de alguns anos para cá (SILBERSCHATZ, KORTH e SUDARSHAN, 1999).

A implantação da informatização dos processos nas empresas tem ocorrido de forma evolutiva e gradual. Inicialmente, apenas determinadas funções eram automatizadas. Mais tarde, à medida que o uso da tecnologia foi se estabelecendo, novas funções foram sendo automatizadas. Dentro dessa perspectiva, surge um problema: a redundância de dados. A redundância de dados ocorre quando a mesma informação está representada em um banco de dados por várias vezes repetidamente (HEUSER, 2004). Existem duas formas nas quais o dado pode apresentar redundância: na forma controlada e na forma não controlada.

A redundância controlada de dados acontece quando o *software* tem conhecimento da múltipla representação da informação e garante a sincronia entre as diversas representações do ponto de vista do usuário externo ao sistema em computador. Essa forma de redundância é utilizada para melhorar a desempenho global do sistema (TANENBAUM, 1995). Se uma mesma informação é armazenada em vários computadores, o acesso torna-se mais rápido, já que pode partir de qualquer um deles.

Já a redundância não controlada de dados acontece quando a responsabilidade pela manutenção da sincronia entre as diversas representações de uma informação está com o usuário e não com o *software* (SOROR *et. al.*, 2009). Este tipo de redundância deve ser evitado, pois traz consigo vários tipos de problemas: (a) repetição na digitação dos dados; e (b) inconsistências dos dados. O resultado é um banco de dados com informações inconsistentes. A solução encontrada para evitar a redundância não controlada de informações é o compartilhamento dos dados. Nessa forma de processamento, cada informação é armazenada uma única vez, sendo acessada pelos vários sistemas que dela necessitam. A FIG. 1 abaixo demonstra uma visão do esquema descrito anteriormente na tentativa de solucionar os problemas de redundância não controlada de dados (DATE, 2000).



Figura 1: Esquema de arquivo de dados unificado

Fonte: Elaborado pelos autores

No próximo tópico serão discutidas as formas de gerenciar os dados por meio de ferramentas tecnológicas capazes de auxiliar na modelagem e gestão de dados em uma organização.

3.3. Sistema Gerenciador de Banco de Dados (SGBD)

3.3.1. Histórico de Utilização

A utilização comercial dos bancos dados teve início nos anos 60. Nessa época, a informação era guardada em arquivos e a sua consulta e manipulação eram processos considerados complexos. A necessidade de utilização desses sistemas surgiu devido ao aumento da quantidade de informação que as empresas passaram a acumular e armazenar. Por não ser possível manter essas informações em memória volátil, era necessário utilizar dispositivos de armazenamento permanentes e de grande capacidade (OLIVEIRA, 2006).

No início dos anos 70 surgiram os bancos de dados relacionais, cuja utilização segue uma curva crescente desde a década de 1970 até os dias de hoje. O sucesso de utilização desse tipo de ferramenta pelas empresas pode ser explicado pela simplicidade no armazenamento e na análise dos dados guardados nesses dispositivos. O modelo relacional

recebeu esse nome por uma característica de constituição: é formado somente por relações entre elementos. Essas relações podem ser mapeadas e trabalhadas por uma linguagem de manipulação chamada de *Structured Query Language* – ou linguagem estruturada de consulta, ou simplesmente SQL (RYS, 2011).

Grandes avanços aconteceram nos últimos 20 anos no que diz respeito às tecnologias relacionadas aos SGBDs, que foram se tornando cada vez mais eficientes, fornecendo inúmeras funcionalidades que melhoraram o acesso dos indivíduos às suas bases de dados, proporcionando maior velocidade, consistência, e segurança no acesso aos dados. Os SGBDs atuais oferecem segurança, *backups* e replicações para recuperação de falhas do equipamento, implementação de restrições, melhoria do tempo de resposta de consultas já realizadas, dentre outros aspectos. Além disso, os SGBDs são sistemas que passaram a executar diversas funções de forma automática, simples e sem interferência do usuário (ANTHES, 2010).

3.3.2. Conceitos e Aplicações

No início da computação, os programas tinham como único objetivo armazenar e manipular dados. Esses programas gravavam seus dados em disco, dentro de estruturas próprias. Programas que não contivessem a estrutura dos dados não poderiam utilizar-se deles. Se diversos desses programas precisassem compartilhar dados de um mesmo arquivo, todos os programas deveriam conhecer e manipular as mesmas estruturas (ELMASRI e NAVATHE, 2005).

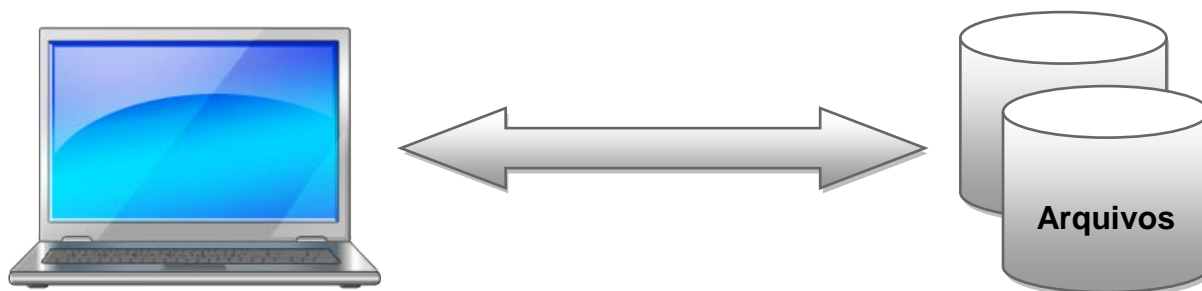


Figura 2: Esquema de arquivos com acessos individuais

Fonte: Elaborado pelos autores

Conforme representa a FIG. 2, a dificuldade em armazenar arquivos individuais para acesso apresenta algumas falhas. A primeira reside na questão de que, se algum programa precisasse realizar alguma mudança na estrutura de dados, todos os programas que acessam esse mesmo arquivo deveriam ser alterados, mesmo que a alteração ocorresse em dados não manipulados por determinados sistemas. Essa ausência de unicidade das estruturas de dados devido à existência de redundâncias é um dos principais problemas existentes em esquemas dessa natureza (ELMASRI e NAVATHE, 2005).

Para evitar essa redundância, Anthes (2010) afirma que a solução encontrada pelos engenheiros da computação foi implementar um sistema intermediário entre o computador e o arquivo. Esse sistema: (a) conhece a estrutura de dados do arquivo; (b) fornece apenas

dados que o programa precisa; e (c) armazena adequadamente os dados de cada programa. Tal esquema é representado pela FIG. 3 a seguir, que expõe a comunicação entre o arquivo e a máquina, tendo o sistema interpretando as necessidades de ambos e fornecendo a melhor alternativa para cada solicitação de dados entre os elementos:

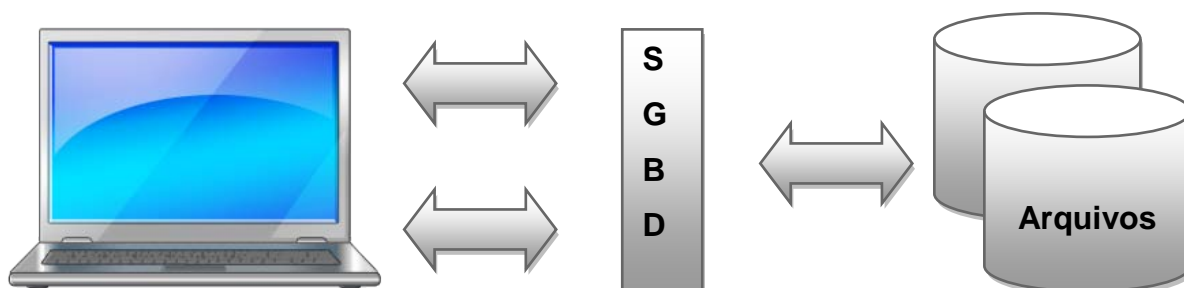


Figura 3: Esquema de arquivo e programa com sistema intermediário

Fonte: Elaborado pelos autores

A partir da implementação desse esquema, os programas são capazes de visualizar apenas os dados que lhes são pertinentes. Tais programas não necessitam conhecer os detalhes de como seus dados estão gravados fisicamente. Dessa forma, não é necessário modificar os sistemas se a estrutura de dados que eles utilizam não for modificada: as alterações ficam concentradas no sistema intermediário. A essa coleção de arquivos foi dado o nome de banco de dados e o sistema intermediário recebeu o nome de sistema gerenciador de banco de dados (SILBERSCHATZ, KORTH e SUDARSHAN, 1999).

Um sistema gerenciador de banco de dados é um sistema capaz de gerir os dados e seus usuários em um ambiente único e consistente. Os SGBDs são capazes de controlar os elementos constituintes de um banco de dados – tabelas, índices, consultas – além de controlar também os relacionamentos e os acessos a cada um desses elementos. Em relação aos seus aspectos constitutivos, os SGBD's comerciais existentes são compostos por três níveis de arquitetura distintos, sendo arquitetura a arte de expressar um modelo ou conceito de informação utilizada em atividades que exigem detalhes explícitos de sistemas complexos, como nos casos dos sistemas de banco de dados, são eles (DATE, 2000):

1. Nível Físico: corresponde à forma como os dados da base de dados são armazenados e organizados internamente no sistema de informação;
2. Nível Conceitual: corresponde à forma como os dados são estruturados ou organizados no banco de dados, juntamente com as relações que existem entre eles. Esse processo consiste na organização da informação em tabelas e relacionamentos e é executado, na maioria das vezes, por profissionais conhecidos como administradores de banco de dados;
3. Nível de Visualização: corresponde à forma como os dados são apresentados aos usuários finais, por meio de interfaces gráficas existentes na aplicação ou nos módulos do SGBD. Este nível é considerado o mais alto da arquitetura da base de dados, pois neste caso, os profissionais que trabalham com o nível de visualização

devem apenas saber trabalhar com as *interfaces*, não sendo necessário conhecer a forma como os dados são armazenados fisicamente, nem na forma como foram organizados conceitualmente.

Do mesmo modo que sua utilização é ampla, é também ampla a gama de modelos de SGBDs existentes. Cada organização deve verificar qual modelo é mais adequado às suas necessidades.

3.3.3. Modelos de SGBDs disponíveis no mercado

Para um melhor aproveitamento das tecnologias existentes na manipulação dos SGBDs, foram idealizados quatro principais modelos que buscam atender às necessidades das organizações. Mardegan *et. al.* (2002) descreve-os da seguinte forma:

- **Hierárquico:** esse modelo de SGBD foi projetado como um sistema que organiza os dados de maneira hierárquica. Dessa forma, para que seja possível acessar um registro na estrutura é preciso navegar entre as hierarquias. Esse modelo exige dos indivíduos que irão manipulá-lo o conhecimento da representação física do banco de dados, conforme demonstra a FIG. 4 abaixo.

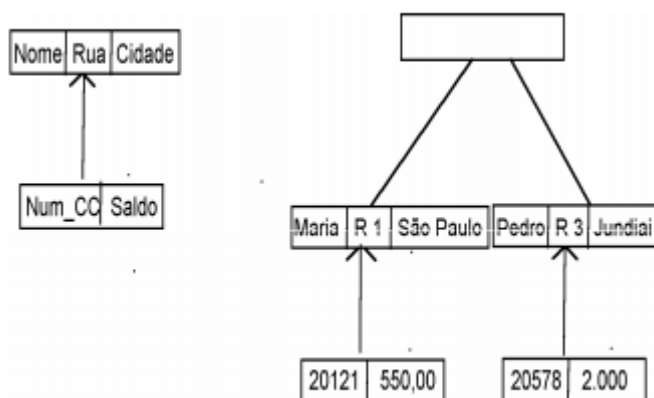


Figura 4: Diagrama Hierárquico

Fonte: Takai, Italiano e Ferreira, 2005: p.7

- **Em Rede:** Sua organização é semelhante à dos BDs hierárquicos. Neste modelo as entidades se representam como nós e suas relações são as linhas que os unem. Nesta estrutura qualquer componente pode se relacionar com qualquer outro. Diferentemente do modelo hierárquico, neste modelo, um filho pode ter vários pais. O Modelo em rede possui uma linguagem padrão para navegar por esta estrutura, conforme apresentado na FIG. 5 abaixo:

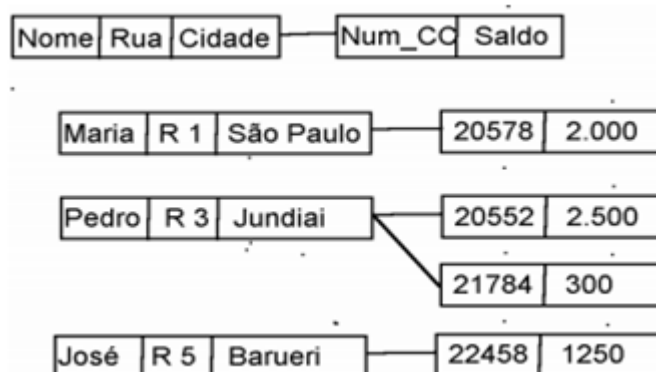


Figura 5: Diagrama em Rede

Fonte: Takai, Italiano e Ferreira, 2005: p.9

- **Relacional:** organiza os dados em uma estrutura de tabela (relação). Para acessar os dados nas tabelas utiliza-se uma linguagem específica, ou linguagem SQL. Não é necessário conhecer a representação física do banco de dados, pois a abordagem relacional separa o sistema aplicativo da implementação física do banco de dados. Dessa forma, mudanças na representação física dos dados não impactam no sistema aplicativo.
- **Orientado a Objetos:** oferece persistência de objetos, incluindo associações e métodos. A falta de um modelo de dados é o principal ponto franco de um SGBDOO. O foco do trabalho é abordar o modelo relacional, não sendo o objetivo desse artigo trabalhar os demais tipos de SGBD's.

3.3.4. Destaque para o modelo relacional

Os primeiros sistemas de banco de dados eram caros e difíceis de utilizar, exigindo especialistas treinados para usar o *software*. Com tal dificuldade, e frente ao crescimento da demanda por esses sistemas. Esse modelo, que domina o mercado de banco de dados, vem se tornando um padrão internacional: é o SGBD relacional. A maioria dos sistemas desenvolvidos hoje em dia utiliza o SGBD relacional como ferramenta de apoio ao acesso a banco de dados. Mesmo nos ambientes de grande porte, os SGBDs relacionais estão crescendo, substituindo os SGBDs de outras abordagens (hierárquico, redes ou sistemas proprietários) (HEUSER, 2004).

Para Kroenke (1999), a descrição da abordagem do SGBD relacional começa com o conceito de tabela. Uma tabela equivale a um conjunto ordenado de linhas ou tuplas, e colunas. Cada linha é composta por uma série de campos, cujos valores são considerados os atributos daquele campo. Cada campo é denominado de forma a representar seu conteúdo. Desse modo, o conjunto de campos que possuem o mesmo nome forma uma coluna. Heuser (2004) argumenta que se compararmos uma tabela de um banco de dados relacional com um arquivo convencional do sistema de arquivos de um computador, são notadas algumas diferenças. Primeiro, as linhas de uma tabela não são ordenadas, isso porque a ordem de recuperação pelo SGBD é aleatória, a menos que uma consulta

especifique explicitamente uma ordenação. As linguagens de consulta permitem o acesso às tabelas por quaisquer critérios que envolvam os campos de uma ou mais linhas. Nos arquivos convencionais, para que você possa buscar registros com base nos valores dos seus campos de forma ágil, normalmente exige-se que haja algum tipo de caminho de acesso. Esse caminho equivale a uma estrutura auxiliar, como um índice ou uma cadeia de ponteiros, que acelera a recuperação de registros por determinados critérios, evitando assim uma leitura exaustiva de todos os registros de um arquivo (HEUSER, 2004).

Para facilitar o acesso aos dados no banco de dados relacional, um conceito básico que estabelece relação entre linhas de tabelas é o conceito de chave. Nesse tipo de sistema, existem ao menos três tipos de chaves a considerar: chave primária, chave estrangeira e chave alternativa. Uma chave primária refere-se a uma coluna ou uma combinação de colunas que contém valores exclusivos, distinguindo cada linha das demais existentes, definindo uma restrição de integridade (DATE, 2000; HEUSER, 2004). Uma chave estrangeira pode ser uma coluna ou uma correlação de colunas, cujos valores aparecem na chave primária da tabela. A chave estrangeira é responsável por implementar os relacionamentos em um banco de dados relacional. Com a existência de uma chave estrangeira, algumas regras serão garantidas em diversas situações de alteração do banco de dados (HEUSER, 2004).

Ainda existem algumas discordâncias acerca do fato de que chave estrangeira sempre referencia uma chave primária de outra tabela. Entretanto, essa restrição não existe, já que uma chave primária pode referenciar a chave primária da própria tabela. Além das chaves primária e secundária, existe também a chamada chave alternativa. Para Date (2000), uma chave alternativa pode servir, em alguns casos, em mais de uma coluna de uma tabela. Em uma estrutura de tabela, normalmente uma coluna é escolhida como chave primária e as demais colunas ou combinações de colunas são consolidadas na chave alternativa.

No que tange o preenchimento dos dados nas tabelas, são muitas as regras que são estabelecidas para essa tarefa. Primeiro, é necessário verificar que, quando criamos uma tabela no banco de dados, para cada coluna desta tabela é necessário especificar um conjunto de valores. A esse conjunto de valores damos o nome de domínio da coluna ou domínio do campo. É necessário então especificar para cada campo da coluna, a possibilidade de estar vazio (*nul*) ou não. Estar vazio indica que não foi atribuído ao campo nenhum valor em seu domínio. As colunas onde não são admitidos valores vazios são chamadas de colunas obrigatórias. As colunas que permitem que os campos tenham valor vazio são chamadas de colunas opcionais. Normalmente, o sistema SGBD relacional exige que todas as colunas primárias sejam preenchidas com valores, porém, para as outras chaves não exigem obrigatoriedade de preenchimento dos campos (HEUSER, 2004).

Dessa forma, é possível considerar que um dos maiores objetivos dos SGBDs é garantir integridade dos dados. Um banco de dados íntegro corresponde à realidade adequadamente representada pelo banco de dados. Para que o SGBD mantenha essa integridade, são utilizados mecanismos de consistência de dados (HEUSER, 2004). As regras que garantem a integridade de qualquer banco de dados devem ser garantidas automaticamente por um SGBD relacional, não sendo exigido de um programador que sejam escritos procedimentos para garantir essa integridade. A essas restrições damos o nome de restrições semânticas (ELMASRI e NAVATHE, 2005). Considerando a otimização de desempenho para os SGBDs, vejamos, no próximo tópico, como os sistemas de banco de dados podem tornar a experiência de acesso aos dados melhor para os usuários do sistema.

3.4. Técnicas de Otimização para os SGBDs

Na implementação de um sistema informação por uma organização, é imprescindível a instalação de um SGBD ligado a esse sistema. Dessa forma, cresce o volume de dados que estes sistemas devem gerenciar e a complexidade de suas aplicações. Tendo em vista esse cenário, realizar operações de forma eficiente e sobre um grande volume de dados é uma questão fundamental para o SGBD, pois a qualidade de seu desempenho está ligada ao tempo de resposta nas operações realizadas no banco de dados (GUNDERLOY e JORDEN, 2001).

Hoje no mercado encontram-se disponíveis diversos SGBDs, que, na maioria dos casos, são instalados, configurados e utilizados em seus valores padrão de fabricação, sem que haja questionamento dos administradores do SGBD sobre o tipo de aplicação em que esse banco de dados estará conectado. Também não é questionado para onde os dados serão encaminhados, seja para um *hardware*, um *software* ou para o sistema operacional. Desse modo, nem sempre é obtido o melhor desempenho possível para o banco de dados, tendo em vista que diversos dos elementos padrão podem ser reconsiderados e ajustados para obtenção de melhor desempenho do SGBD, quando conectado a uma aplicação (GUNDERLOY e JORDEN, 2001). Para garantir o melhor desempenho nas consultas feitas ao banco de dados, vejamos, no próximo tópico, algumas técnicas que possibilitam essa melhoria.

3.4.1. Consultas com melhor desempenho

O monitoramento e o ajuste de consultas SQL é a atividade que consome grande tempo de administradores de banco de dados, devido à complexidade envolvida nessa tarefa. Muitas dessas consultas não alcançam um desempenho satisfatório, pois são escritas objetivando o resultado imediato, e não o melhor caminho para obtê-lo. Fatores como falta de experiência em desenvolvimento, baixo nível de conhecimento técnico, prazos de entregas de projetos subdimensionados e falta de monitoramento do banco de dados contribuem para que as consultas sejam ineficientes (ELMASRI e NAVATHE, 2005).

A partir desse cenário, o SGBD constrói o plano de execução a partir da consulta SQL criada, tendo uma margem limitada de opções sobre os operadores utilizados. Uma consulta SQL mal escrita leva o SGBD a utilizar um caminho que nem sempre é o mais adequado, o que gera um plano de execução que normalmente compromete o desempenho. Em ambientes nos quais existem inúmeras consultas construídas dessa forma, a consequência acaba sendo a dificuldade no uso eficiente da maior parte dos recursos disponíveis, deixando, muitas vezes, de atender em tempo hábil outros processos críticos cujas execuções são rápidas e necessitam de prioridade (ELMASRI e NAVATHE, 2005).

Existem algumas técnicas utilizadas pelos administradores de banco de dados a fim de alcançar melhor desempenho nessas atividades. A justificativa sobre a utilização da técnica de reescrita de consultas reside no fato de que essa técnica afeta apenas uma consulta específica, não se propagando para outras aplicações que acessam as tabelas envolvidas na instrução SQL (ELMASRI e NAVATHE, 2005). Com isso, alguns critérios são adotados na tentativa de otimizar o desempenho de consultas SQL, que serão explorados no próximo tópico.

3.4.2. Critérios adotados para buscar otimização em consultas SQL

De uma forma simplificada, existem alguns critérios principais que podem ser adotados para identificar as consultas que devem ser modificadas no banco de dados. São eles:

1. Monitorar as sessões ativas que estão sendo executadas no banco de dados;
2. Separar as consultas que estão com execuções lentas;
3. Dividi-las em grupos, como: prioridade, frequência de execução e baixo desempenho;
4. Implementar os ajustes reescrevendo as consultas que estão com desempenho baixo.

Cada banco de dados fornece ferramentas específicas para capturar as consultas, porém a maneira mais eficiente e explícita é o próprio uso feito pelo usuário final (BATTISTI, 2005; RIBEIRO, 2009). Outra técnica possível para otimizar as consultas é fazê-lo por meio de índices.

3.4.3. Melhorando o desempenho com a implementação de índices

Segundo Ribeiro (2009), índices são estruturas que possuem algoritmos otimizados para acessar dados em um banco de dados. Tais estruturas constituem uma ferramenta poderosa para o projetista do Banco de Dados no intuito de auxiliá-lo na melhoria do desempenho das consultas. O principal objetivo de um índice é reduzir o número de operações de I/O (leitura e escrita) necessários para localizar os dados solicitados por uma requisição.

Sendo assim, ao fazer uso dos índices, o SGBD rapidamente localizará e disponibilizará os dados de uma consulta por meio de um número de I/O muito menor. Se não utilizar índices, o SGBD precisará realizar uma operação conhecida como *Table Scan*, para localizar os dados solicitados. Uma operação *Table Scan* é uma leitura sequencial de todos os registros da tabela, o que implica muito mais operações de I/O no disco. Essas operações são normalmente lentas quando comparadas com operações de leitura e escrita em memória (BATTISTI, 2005).

Os índices podem ser criados em qualquer coluna da tabela, inclusive em uma coluna com valores calculados. O corpo de um índice é formado pelas colunas de uma tabela cujos dados se deseja classificar seguido de uma referência conhecida como “ponteiro”, que serve para localizar a página de dados da tabela (BATTISTI, 2005). Uma vez tendo verificado alguns dos tipos mais comuns de otimização de consultas, será descrito o principal meio de otimização objetivado por esse artigo: o particionamento horizontal de tabelas.

3.4.4. A otimização por meio do particionamento horizontal

O conceito de particionamento não é novo para os SGBDs. As formas de particionamento surgiram em cada nova atualização desses sistemas. No entanto, sem recursos especificamente projetados para ajudar o usuário a criar e manter um esquema de

particionamento, os bancos de dados têm tido baixo desempenho de utilização. Além disso, os usuários e desenvolvedores podem não compreender da melhor forma o esquema de particionamento (devido a um projeto de banco de dados mais complexo), tornando reduzidos os benefícios dessa implementação (ELMASRI e NAVATHE, 2005).

O particionamento em SGBDs é utilizado para melhorar o desempenho e o gerenciamento de grandes tabelas, ou tabelas que apresentem diferentes padrões de acesso. Normalmente, os usuários criam tabelas para armazenar informações sobre uma entidade, como clientes ou vendas, e em cada tabela existem atributos que descrevem apenas uma entidade. Essas tabelas não são necessariamente otimizadas por padrão, e na medida em que a tabela cresce, mais lento será o desempenho e mais difícil será o gerenciamento dessa tabela.

Enquanto o tamanho de um banco de dados muito grande (SGBD) é medido em centenas de *gigabytes* ou *terabytes*, o termo não indica necessariamente o tamanho real das tabelas individuais dentro do banco de dados. Um banco de dados grande é aquele que não apresenta o desempenho desejado ou aquele em que os custos operacionais extrapolaram os custos de manutenção predefinidos inicialmente para aquele sistema. Essas exigências também se aplicam a tabelas. Uma tabela pode ser considerada grande se as atividades de outros usuários ou operações de manutenção têm um efeito limitante na disponibilidade de utilização da tabela. Na técnica de particionamento de tabelas, é mais importante verificar o tamanho da tabela, pois é nela que será feito o filtro limitante desejável, a fim de se obter um melhor desempenho no acesso aos dados (QIAN *et. al.*, 2006).

Além do tamanho, uma tabela com diferentes padrões de acesso pode ser uma preocupação para o desempenho e disponibilidade de acesso. Embora os padrões de uso nem sempre variem, quando isso ocorre, o particionamento pode resultar em ganhos adicionais de gestão, desempenho e disponibilidade. Além disso, quando grandes conjuntos de dados estão sendo usados de maneiras diferentes, as operações de manutenção frequentes são executadas em dados estáticos. Isto pode ter efeitos caros, problemas de desempenho, problemas de bloqueio, problemas de *backup* (custo de espaço, tempo e operação), bem como um impacto negativo sobre a disponibilidade global do servidor (ELMASRI e NAVATHE, 2005).

Quando as tabelas e índices se tornam muito grandes, o particionamento pode ser útil a partir da divisão dos dados em seções menores e mais facilmente gerenciáveis. A definição do conjunto de particionado pode ser personalizado, definido e gerenciado a partir do entendimento sobre a necessidade em se aplicar essa técnica. Dessa forma, se existe uma tabela grande em um sistema com múltiplos processadores, o particionamento da tabela pode levar a um melhor desempenho, que acontece por meio de operações no banco de dados que ocorrem de forma paralela. O desempenho de operações em conjuntos de dados extremamente grandes (por exemplo, milhões de linhas) pode ser mais lento se comparado à realização de várias operações em subconjuntos individuais de dados, acessados de forma paralela (QIAN *et. al.*, 2006). Uma das técnicas que são capazes de promover o particionamento em tabelas de dados é denominada particionamento horizontal.

O particionamento horizontal divide uma tabela em várias tabelas. Cada tabela contém o mesmo número de colunas, mas menos linhas. Uma tabela que contenha um bilhão de linhas pode ser particionada horizontalmente em doze tabelas, sendo cada tabela representando, por exemplo, um mês de dados de um ano específico. Qualquer consulta, ao requerer dados de um mês específico, fará referência somente à tabela correspondente. Determinar como particionar as tabelas horizontalmente depende de como os dados são analisados. É necessário particionar o banco de dados de forma que as consultas façam

referência ao menor número possível de tabelas. Por exemplo, uma tabela pode conter dados dos últimos cinco anos, mas somente os dados do ano atual são acessados regularmente. Nesse caso, é possível considerar o particionamento dos dados em cinco tabelas, com cada tabela contendo dados somente de um ano.

5. Metodologia

Nesse capítulo, apresenta-se a metodologia da pesquisa, que reúne o tipo e a técnica da pesquisa, a unidade de análise, além da descrição do processo de coleta, tratamento e análise dos dados.

5.1. Tipo de Pesquisa

Neste estudo, optou-se pela realização desta investigação por meio de uma metodologia de natureza quantitativa. A pesquisa quantitativa pode ser definida como um método de pesquisa social que utiliza técnicas estatísticas e metodologia de pesquisa estruturada. Tem como objetivo quantificar dados e generalizar os resultados da amostra para o universo especificado. Serve especialmente para gerar medidas precisas e confiáveis que permitam uma análise estatística. É apropriada para medir tanto opiniões, atitudes e preferências, como comportamentos. Dessa forma, são muitas as maneiras de se fazer a coleta de dados em uma abordagem quantitativa (GUNTHER, 2006). Nesta pesquisa, foram utilizadas:

1. Pesquisa Bibliográfica;
2. Pesquisa documental sobre o processo estudado;
3. Observações diretas do pesquisador nas áreas que compõe a organização;
4. Medições na ferramenta SQL Server por meio de variáveis pré-estabelecidas.

5.2. Técnicas de pesquisa

A fim de validar os conceitos descritos no referencial teórico, decidiu-se utilizar o método de estudo de caso. Para Yin (2005), um estudo de caso é uma investigação que analisa um fenômeno contemporâneo dentro do contexto da vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos. O mesmo autor cita que a investigação do estudo de caso enfrenta uma situação única e se beneficia do desenvolvimento prévio de proposições teóricas para conduzir a coleta e análise dos dados. Como resultado é possível avaliar várias fontes de evidências que comparavam a melhoria após a aplicação do método citado. O estudo de caso tem se tornado a estratégia preferida quando há pouca possibilidade de controle sobre os eventos estudados ou quando o foco de interesse se faz sobre fenômenos atuais, que somente poderão ser analisados dentro desse contexto de vida real (YIN, 2005).

Para esta pesquisa optou-se por realizar um estudo de caso único. Para o estudo de caso, a teoria especifica um conjunto claro de proposições, assim como as circunstâncias

nas quais se acredita que as proposições sejam verdadeiras. Para confirmar, contestar ou estender a teoria, o caso único pode ser utilizado para determinar se as proposições estão corretas ou se algum outro conjunto de explicações poderia ser mais relevante (YIN, 2005).

5.3. Unidade de análise

Utilizou-se uma empresa de consultoria no ramo de tecnologia da informação, que presta serviço de tecnologia para diversas empresas, de diversos ramos de atividade. Neste artigo, a organização onde o caso foi aplicado será denominada empresa *ALFA*, que solicitou sigilo, e requer confidencialidade no conteúdo levantado.

5.4. Coleta de dados

No sentido de contextualizar os motivos e especificidades na adoção do particionamento horizontal em SGBDs, coletou-se um *backup* do banco de dados utilizado na empresa *ALFA*, com grande quantidade de dados armazenados (aproximadamente 70GB). O SGBD utilizado foi o SQL Server versão 2008 que é acessado por cerca de 1800 operadores e que possui aproximadamente 120 contas de usuários com possibilidade de acesso simultâneo a essa base de dados. Foram medidos os seguintes processos internos: acompanhamento da produção, qualidade, exportação dos dados, relatórios, e confirmação de produção (dados primários).

Também foram utilizados dados da Intranet e do site público da empresa *ALFA*. A coleta de dados teve como intuito formar a base para melhor compreensão das razões que levaram a adoção da técnica de particionamento horizontal, bem como a evidência das etapas e reflexos na gestão organizacional. Além disso, foram utilizados manuais de procedimentos, documentos normativos e relatórios de execução do programa que controla a produção na empresa *ALFA* (dados secundários).

5.5. Tratamento e análise dos dados

O tratamento dos dados primários foi feito por meio da aplicação do método de particionamento horizontal em tabelas. Foram gerados gráficos comparativos, estratificados por processos de negócio. O diagnóstico dos dados foi feito a partir do conteúdo do banco de dados, considerando as tabelas que possuem maior quantidade de registros, maior número de acessos e consultas mais longas, além de atividades nas quais os operadores da empresa *ALFA* encontram lentidão no acesso. A análise dos dados da pesquisa teve em vista buscar a interação entre o material teórico pesquisado, a análise documental e as informações medidas nas consultas realizadas, bem como a verificação de elementos que confirmassem ou contradissem os pressupostos teóricos deste trabalho.

Abaixo, a TABELA 1 apresenta o conjunto de variáveis definidas, juntamente com o processo do qual a variável pertence, o tamanho de cada uma das tabelas, a quantidade de registros, o tempo gasto para realizar a consulta considerando tabelas consistidas e não consistidas, além do número de particionamentos a que as tabelas foram submetidas:

Tabela 1: Definição de variáveis de medida de particionamento

Definicao variaveis de medida do particionamento horizontal						
Variaveis	Processo	Tamanho das tabelas	Quantidade de registros	Tempo Sem Consistencia (em s)	Tempo Com Consistencia (em s)	No de particionamentos
V1: PRODUTO_LOTE_*_SEM_PARTICIONAMENTO	Producao	10GB	1300078	150	30	1
V2: PRODUTO_LOTE_*_COM_PARTICIONAMENTO	Producao	10GB	1300078	110	20	8
V3: QUALIDADE_AMOSTRA_*_SEM_PARTICIONAMENTO	Qualidade	8GB	295419	176	37	1
V4: QUALIDADE_AMOSTRA_*_COM_PARTICIONAMENTO	Qualidade	8GB	295419	103	33	8
V5: DADOS_INTEGRADOS_*_SEM_PARTICIONAMENTO	Integracao de Dados	11GB	1085599	195	48	1
V6: DADOS_INTEGRADOS_*_COM_PARTICIONAMENTO	Integracao de Dados	11GB	1085599	183	36	8
V7: RELATORIO_ACOMPANHAMENTO_*_SEM_PARTICIONAMENTO	Relatorio	8GB	37272	25	3	1
V8: RELATORIO_ACOMPANHAMENTO_*_COM_PARTICIONAMENTO	Relatorio	8GB	37272	20	2	8
V9: EXPORTA_ERP_*_SEM_PARTICIONAMENTO	Confirmacao de Producao	15GB	181746	39	31	1
V10: EXPORTA_ERP_*_COM_PARTICIONAMENTO	Confirmacao de Producao	15GB	181746	31	29	10

Fonte: Elaborado pelos autores

As análises decorrentes do particionamento horizontal das tabelas são feitas no próximo tópico, e consideram a verificação da validade dos argumentos apresentados no referencial teórico.

6. Resultados

Os eixos desta pesquisa se baseiam na análise do desempenho e nas perspectivas de utilização do particionamento horizontal em sistemas de gerenciamento de banco de dados. O objetivo desse tópico é demonstrar como o particionamento horizontal pode impactar no desempenho ao acessar os dados por meio de consultas estruturadas (*queries*). De acordo com Qian *et. al.* (2006), o particionamento melhora o gerenciamento de grandes tabelas, possibilitando o acesso e a gestão de conjuntos de dados com melhor desempenho, da mesma forma que sustenta a integridade dos dados, diminui a sobrecarga de I/O dos discos e também pode ser utilizado para otimizar *backups*. Para medir o desempenho das consultas, foram definidas 10 variáveis em dois cenários: sem e com consistência. A consistência é obtida a partir da seleção dos campos mais utilizados pelos usuários dos processos de negócio.

Essas variáveis serão utilizadas para nortear o processo de medição comparativa de desempenho, integridade e confiabilidade. A partir das medições será possível analisar vantagens e desvantagens que o particionamento horizontal pode apresentar. As variáveis escolhidas representam consultas aos processos acessados por grande parte da linha de produção da empresa ALFA, em ações executadas pelos operadores do sistema suportado por essa base de dados. São elas:

6.1. Processo de Produção

Cenário 1: sem consistência:

V1: PRODUTO_LOTE_*_SEM_PARTICIONAMENTO: lotes existentes na produção, no período de 2009 até 2011, considerando todos os campos da tabela não particionada.

V2: PRODUTO_LOTE_*_COM_PARTICIONAMENTO: lotes existentes na produção, no período de 2009 até 2011, considerando todos os campos da tabela particionada.

Cenário 2: com consistência:

V1: PRODUTO_LOTE_*_SEM_PARTICIONAMENTO: lotes existentes na produção, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela não particionada.

V2: PRODUTO_LOTE_*_COM_PARTICIONAMENTO: lotes existentes na produção, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela particionada.

Quantidade de Registros: 1.370.078

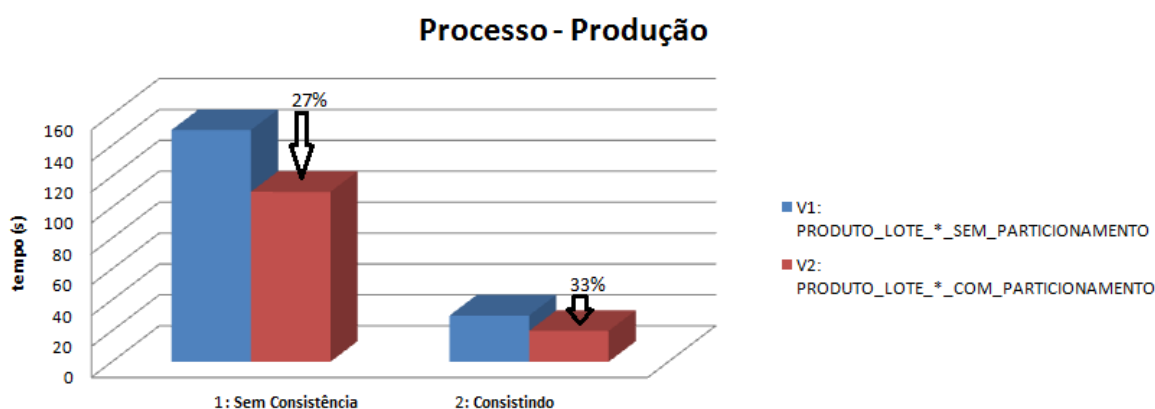


Figura 6: Resultados obtidos – Processo de Produção
Fonte: Dados da pesquisa

A FIG. 6 acima demonstra as melhorias alcançadas nos dois processos – com e sem consistência, após a execução do particionamento horizontal. No primeiro caso (sem consistência), foram conseguidos 27% de redução de tempo, o que é uma melhoria significativa de desempenho. No segundo caso, já com a tabela particionada, conseguiu-se 33% de melhoria de tempo, que irá refletir de forma bastante positiva nas consultas dos clientes desse processo.

6.2. Processo de Qualidade

Cenário 1: sem consistência:

V3: QUALIDADE_AMOSTRA_*_SEM_PARTICIONAMENTO: amostras realizadas em todos os lotes da produção, no período de 2009 até 2011, considerando todos os campos da tabela particionada.

V4: QUALIDADE_AMOSTRA *_COM_PARTICIONAMENTO: amostras realizadas em todos os lotes da produção, no período de 2009 até 2011, considerando todos os campos da tabela particionada.

Cenário 2: com consistência:

V3: QUALIDADE_AMOSTRA *_SEM_PARTICIONAMENTO: amostras realizadas em todos os lotes da produção, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela particionada.

V4: QUALIDADE_AMOSTRA *_COM_PARTICIONAMENTO: amostras realizadas em todos os lotes da produção, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela particionada.

Quantidade de Registros: 295.419

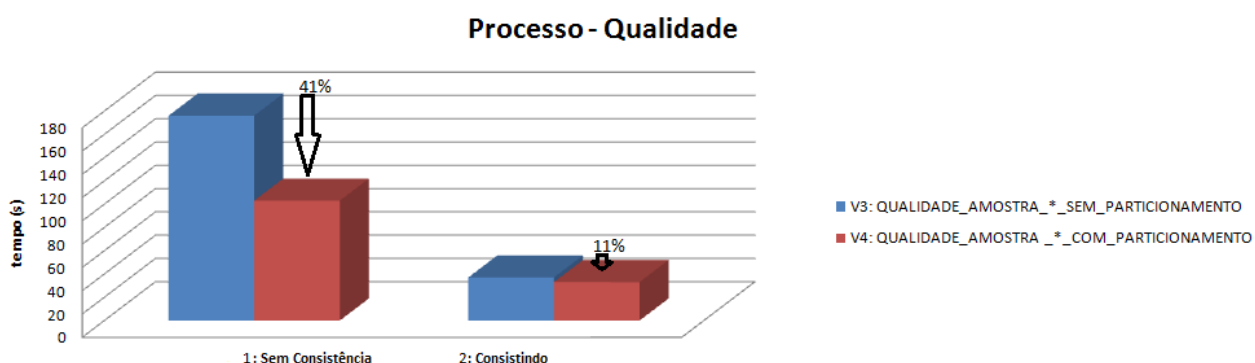


Figura 7: Resultados obtidos – Processo de Qualidade

Fonte: Dados da pesquisa

A FIG. 7 acima demonstra as melhorias alcançadas nos dois processos – com e sem consistência, após a execução do particionamento horizontal. No primeiro caso (sem consistência), foram conseguidos 41% de redução de tempo, maior ganho encontrado, o que é uma melhoria muito significativa de desempenho. No segundo caso, já com a tabela particionada, conseguiu-se 11% de melhoria de tempo, que irá refletir de forma bastante positiva nas consultas dos clientes desse processo.

6.3. Processo de Integração de Dados

Cenário 1: sem consistência:

V5: DADOS_INTEGRADOS *_SEM_PARTICIONAMENTO: dados integrados com o chão de fábrica no período de 2009 até 2011, considerando todos os campos da tabela não particionada.

V6: DADOS_INTEGRADOS *_COM_PARTICIONAMENTO: dados integrados com o chão de fábrica no período de 2009 até 2011, considerando todos os campos da tabela particionada.

Cenário 2: com consistência:

V5: DADOS_INTEGRADOS_*_SEM_PARTICIONAMENTO: dados integrados com o chão de fábrica no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela não particionada.

V6: DADOS_INTEGRADOS_*_COM_PARTICIONAMENTO: dados integrados com o chão de fábrica no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela particionada.

Quantidade de Registros: 1.085.599

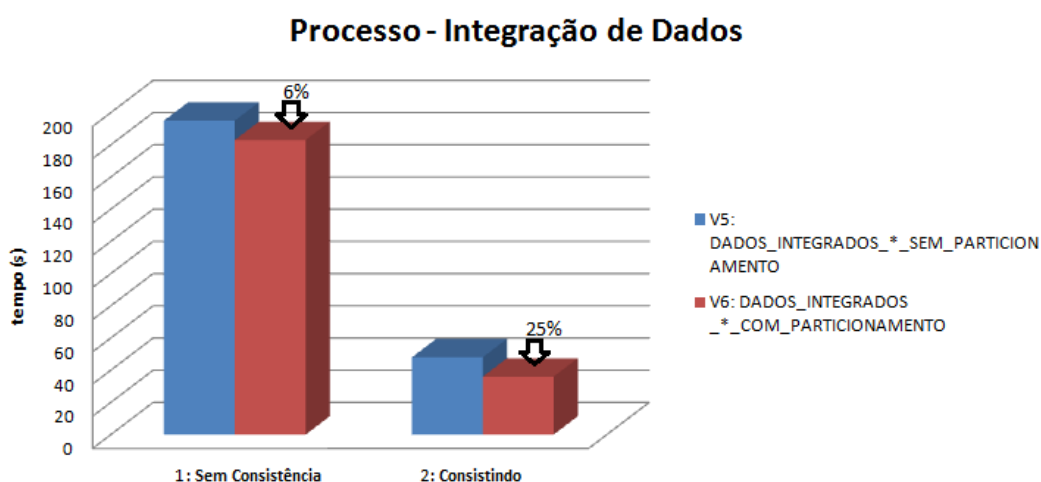


Figura 8: Resultados obtidos – Processo de Integração de dados
Fonte: Dados da pesquisa

A FIG. 8 acima demonstra as melhorias alcançadas nos dois processos – com e sem consistência, após a execução do particionamento horizontal. No primeiro caso (sem consistência), foram conseguidos 6% de redução de tempo, o que é uma melhoria significativa de desempenho. No segundo caso, já com a tabela particionada, conseguiu-se 25% de melhoria de tempo, que irá refletir de forma bastante positiva nas consultas dos clientes desse processo.

6.4. Processo de montagem de Relatórios

Cenário 1: sem consistência:

V7: RELATORIO_ACOMPANHAMENTO_*_SEM_PARTICIONAMENTO: Relatório de acompanhamento da produção, no período de 2009 até 2011, considerando todos os campos da tabela não particionada.

V8: RELATORIO_ACOMPANHAMENTO_*_COM_PARTICIONAMENTO: Relatório de acompanhamento da produção, no período de 2009 até 2011, considerando todos os campos da tabela particionada.

Cenário 2: com consistência:

V7: RELATORIO_ACOMPANHAMENTO_*_SEM_PARTICIONAMENTO: Relatório de acompanhamento da produção, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela não particionada.

V8: RELATORIO_ACOMPANHAMENTO_*_COM_PARTICIONAMENTO: Relatório de acompanhamento da produção, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela particionada.

Quantidade de Registros: 37.272

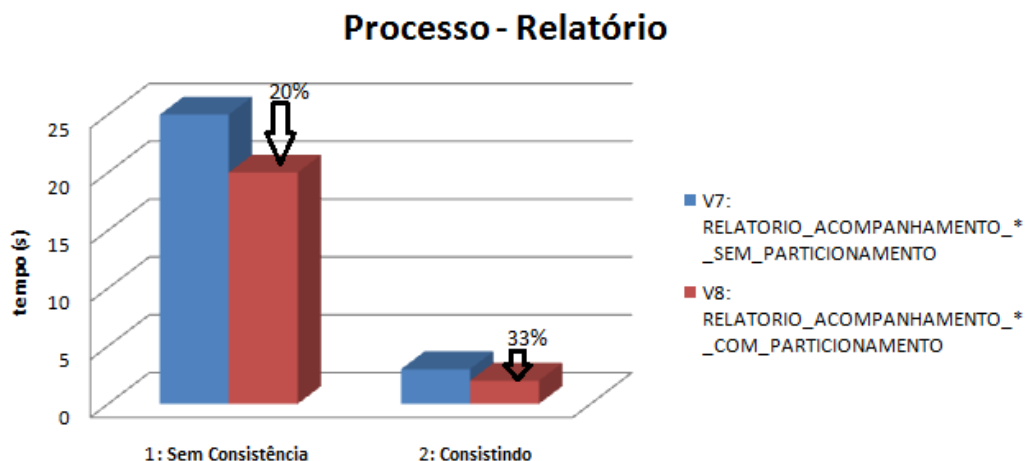


Figura 9: Resultados obtidos – Relatórios

Fonte: Dados da pesquisa

A FIG. 9 acima demonstra as melhorias alcançadas nos dois processos – com e sem consistência, após a execução do particionamento horizontal. No primeiro caso (sem consistência), foram conseguidos 20% de redução de tempo, o que é uma melhoria significativa de desempenho. No segundo caso, já com a tabela particionada, conseguiu-se 33% de melhoria de tempo, que irá refletir de forma bastante positiva nas consultas dos clientes desse processo.

6.5. Processo de Exportação ERP

Cenário 1: sem consistência:

V9: EXPORTA_ERP_*_SEM_PARTICIONAMENTO: dados exportados para o ERP da empresa ALFA, no período de 2009 até 2011, considerando todos os campos da tabela não particionada.

V10: EXPORTA_ERP_*_COM_PARTICIONAMENTO: dados exportados para o ERP da empresa ALFA, no período de 1998 até 2011, considerando todos os campos da tabela particionada.

Cenário 2: com consistência:

V9: EXPORTA_ERP*_SEM_PARTICIONAMENTO: dados exportados para o ERP da empresa ALFA, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela não particionada.

V10: EXPORTA_ERP*_COM_PARTICIONAMENTO: dados exportados para o ERP da empresa ALFA, no período de julho de 2011 até agosto de 2011, considerando todos os campos da tabela particionada.

Quantidade de Registros: 181.746

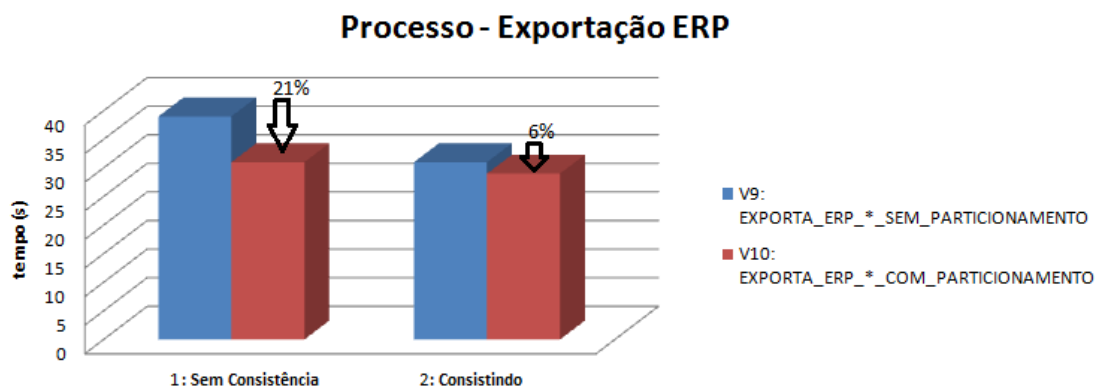


Figura 10: Resultados obtidos – Processo de Exportação ERP
Fonte: Dados da pesquisa

A FIG. 10 acima demonstra as melhorias alcançadas nos dois processos – com e sem consistência, após a execução do particionamento horizontal. No primeiro caso (sem consistência), foram conseguidos 21% de redução de tempo, o que é uma melhoria significativa de desempenho. No segundo caso, já com a tabela particionada, conseguiu-se 6% de melhoria de tempo, que irá refletir de forma bastante positiva nas consultas dos clientes desse processo.

7. Conclusões e Recomendações

O objetivo dessa pesquisa foi analisar o desempenho na resposta à implementação do particionamento horizontal em tabelas considerando um SGBD instalado em uma empresa multinacional no ramo de controle de automação siderúrgico. Para completo entendimento dos processos analisados, procurou-se também: (a) conceituar e contextualizar os sistemas de banco de dados; (b) detalhar o método de particionamento horizontal em tabelas; e (c) analisar as vantagens e desvantagens na utilização do particionamento horizontal em tabelas.

O método de particionamento facilita o gerenciamento de grandes tabelas e / ou índices, permitindo o acesso e o gerenciamento de subconjuntos de dados de forma rápida e eficaz, ao mesmo tempo em que mantém a integridade geral da coleção de dados. Com o particionamento, uma operação como o carregamento de dados de um sistema, leva apenas segundos em vez dos minutos ou horas necessários nas versões anteriores do Microsoft SQL Server ®. Dessa forma, a grande contribuição desse trabalho reside na melhoria do desempenho das consultas executadas pelos operadores, o que pode trazer para os

indivíduos que trabalham com essas tarefas tempo para executar outras tarefas, fazendo-os mais produtivos.

É também considerado um benefício da prática de particionar tabelas a otimização das operações de manutenção realizadas nos subconjuntos de dados particionados – tais operações também são executadas com mais eficiência porque se referem apenas aos dados necessários, sem considerar a totalidade da tabela. A decisão entre implementar ou não implementar o particionamento horizontal em tabelas depende basicamente do tamanho da mesma, além da análise de como ela será utilizada e como será seu desempenho com relação às consultas dos usuários e às operações de manutenção.

Em geral, o particionamento de uma tabela extensa é apropriado quando se aplicam condições como: (1) a tabela contém, ou se espera-se que contenha, muitos dados usados de diferentes maneiras; (2) o desempenho das consultas ou atualizações feitas na tabela não é o esperado ou o custo de manutenção excede os períodos de manutenção predefinidos inicialmente. Operações como essas, normalmente, levam segundos em vez de minutos ou horas que demoravam nas versões anteriores.

O particionamento de uma tabela ou índice pode aprimorar o desempenho das consultas se as partições forem projetadas corretamente, com base no tipo de consultas executadas com frequência em sua configuração de *hardware*. No método de particionamento horizontal visto neste estudo, uma tabela é dividida em várias outras tabelas. Cada tabela contém o mesmo número de colunas, mas reduzem-se as linhas. Uma tabela que contém um milhão de linhas pode ser particionada horizontalmente em doze tabelas, sendo cada tabela menor representando um mês de dados de um ano específico. Qualquer consulta, ao solicitar dados de um determinado mês, fará referência somente àquela tabela apropriada. Determinar como particionar as tabelas horizontalmente depende de como os dados são analisados. É necessário, portanto, particionar as tabelas de forma que as consultas executadas com maior frequência façam referência ao menor número possível de tabelas. Caso contrário, consultas desse tipo, feitas em excesso, usadas para mesclar as tabelas logicamente no momento da consulta, podem afetar negativamente o desempenho.

Conforme constatado na pesquisa, verifica-se que a aplicação do particionamento horizontal em um SGBD é válida, desde que se tenha grande quantidade de dados e espaço em disco disponível, pois foi verificado um aumento considerável na utilização de espaço em disco. Verificou-se também que o particionamento deve ser feito preferencialmente no início da modelagem do banco de dados, a fim de limitar os campos reais das consultas mais demoradas e já planejar um futuro particionamento em consultas que possam originar algum gargalo de processamento no futuro. Os ganhos de desempenho no banco de dados foram notáveis após o particionamento, mas foi verificado também que o particionamento de uma tabela deve ser estudado antes de ser executado, para que não haja aplicação em uma tabela que contenha poucos dados e assim o desempenho piore.

Mesmo assim, esse assunto não se esgota aqui. Algumas pesquisas podem ser construídas ao avaliar este estudo, a fim de medir qual método deve-se aplicar de acordo com modelo de dados como, por exemplo, o particionamento vertical. O particionamento vertical divide uma tabela em várias outras que contêm menos colunas. Também é possível estudar a aplicação do método de particionamento horizontal em outros tipos de SGBDs como *Oracle*, *Firebird*, *PostgreSQL* e saber se há melhora ou piora e se existirão maiores ganhos em aplicar o método de particionamento horizontal em um destes SGBDs.

Essa pesquisa também apresenta algumas limitações. A primeira delas é o espaço em disco disponível no ambiente, que deve estar dimensionado para acomodar o novo volume de dados gerado. Além disso, a falta de acessos simultâneos, contrário ao que

acontece na empresa ALFA, caracteriza outra limitação dessa pesquisa, já que vários usuários realizam diversas consultas simultâneas. Esse processo de acessos simultâneos acarreta em lentidão no sistema em produção. Uma última limitação verificada diz respeito à inexistência de acompanhamento do processo de elaboração das consultas pelos operadores da área de produção da empresa ALFA, que poderiam auxiliar no levantamento de outros processos essenciais para a organização, que não foram abordados nessa pesquisa.

Referências

ABRAHAM, S., HENRY, F. K., SUDARSHAN, S. **Sistemas de Banco de Dados**. 3 ed. São Paulo: São Paulo, 1999.

ALBERTO, C. H. **Projeto de Banco de Dados**. 4 ed. Rio Grande do Sul: Porto alegre, 2008.

ANGELONI, M. T. Elementos intervenientes da tomada de decisão. **Ci. Inf., Brasília**, v. 32, n. 1, p. 17-22, jan./abr. 2003.

ANTHES, G. Happy Birthday, RDBMS! **Communications of the ACM**. Vol. 53. N. 5. Maio 2010.

BATTISTI, J. **SQL Server 2005: Administração e Desenvolvimento: Curso Completo**. Rio de Janeiro: Axcel Books, 2005.

CASTELLS, Manuel. **Sociedade em Rede: A era da informação. Economia, sociedade e cultura**, v1. São Paulo: São Paulo, 1999.

CUNHA, C.B. **Uma contribuição para o problema de roteirização de veículos com restrições operacionais**. São Paulo: EPUSP, Departamento de Engenharia de Transportes. 222p. (Tese de Doutorado). 1997.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. 7 ed. Rio de Janeiro: Campus, 2000.

ELMASRI, R.; NAVATHE, S. B. [Trad.]. **Sistemas de bancos de dados**. Traduzido do original: FUNDAMENTALS OF DATABASE SYSTEMS. São Paulo: Pearson (Addison Wesley), 2005. 724 p.

GUNDERLOY, M.; JORDEN, J. L. **Dominando o SQL Server 2000**. São Paulo : Makron Books, 2001.

GUNTHER, H. Pesquisa Qualitativa Versus Pesquisa Quantitativa: Esta É a Questão? **Psicologia: Teoria e Pesquisa**. Mai-Ago 2006, Vol. 22 n. 2, pp. 201-210.

HEUSER, C. A. **Projeto de Banco de Dados**. Série Livros Didáticos – Instituto de Informática da UFRGS: Ed. Sagra-Luzzatto. 5 ed. 2004.

KROENKE, D. M. **Banco de Dados: Fundamentos, Projeto e Implementação**. 6ª. Ed. Rio de Janeiro: LTC, 1999.

MARDEGAN, R., AZEVEDO, R.C., OLIVEIRA, J.F.G. Os Benefícios da Coleta de Dados do Chão-de-fábrica para o Processo de negócio Gestão da Demanda. **Proceedings of 22th Encontro Nacional de Engenharia de Produção**, Curitiba, v. 01. p. 01-08, 2002.

OLIVEIRA, A. C. J. **Implantação de Sistema MES para gestão da produção**. Graduação em Sistemas de Informação: Uberlândia / Minas Gerais. Monografia. 2008.

OLIVEIRA, A. P. **O Modelo de replicação de dados entre SGBD Heterogêneos**. Universidade Luterana do Brasil. Graduação em Ciência da Computação: Campinas / São Paulo. Monografia. 2006.

QIAN, G.; ZHU, Q.; XUE, Q.; PRAMANIK, S. Dynamic Indexing for Multidimensional Non-ordered Discrete Data Spaces Using a Data-Partitioning Approach. **ACM Transactions on Database Systems**, Vol. 31, No. 2, Jun. 2006, P. 439–484.

RABELO, E. **Avaliação de técnicas de visualização para mineração de dados**. Maringá: UEM, Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual do Maringá / Paraná, Dissertação de Mestrado. 2007.

RIBEIRO, P. SQL Magazine: **Entendendo e utilizando índices na otimização de queries no SQL Server**. São Paulo: São Paulo, 2009.

ROSINI, A. M.; PALMISANO, A. **Administração de Sistemas de Informação e a Gestão do Conhecimento**. São Paulo: Thomson, 2003.

RYS, M. Scalable SQL. **Communications of the ACM**. Vol. 54. N. 6. Jun. 2011.

SELTZER, M. Beyond Relational Databases. **Communications of the ACM**. Vol. 51. N. 7. Jul. 2008.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de banco de dados**. 3. ed. São Paulo: Makron Books, 1999.

SILVA, S. L. Gestão do Conhecimento: uma revisão crítica orientada pela abordagem da criação do conhecimento. **Ciência da Informação**, Brasília, v. 33, n. 2, p. 143-151, maio/ago. 2004.

SOROR, A. A.; MINHAS, U. F.; ABOULNAGA, A.; SALEM, K.; KOKOSIELIS, P.; KAMATH, S. Automatic Virtual Machine Configuration for Database Workloads. **ACM Transactions on Database Systems**, Vol. 35, No. 1, Article 7, Fev. 2010.

SOUKUP, R.; DELANEY, K. **Desvendando o Microsoft SQL Server 7.0**. Rio de Janeiro: Campus, 1999.

STONEBRAKER, M.; CATTELL R. Conceptually Modeling Windows and Bounds for Space and Time in Database Constraints. **Communications of the ACM**. Vol. 51. N. 11. Nov. 2008.

TAKAI, O. K.; ITALIANO, I. C.; FERREIRA, J. E. **Introdução ao banco de dados (apostila)**. Disponível em: <http://www.ime.usp.br/~jef/apostila.pdf>. Acesso em: 20/12/2011.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**, Rio de Janeiro: Prentice-Hall do Brasil Ltda, 1995.

WHALEN, E., GARCIA, M., PATEL, B., MISNER, S., ISAKOV, V. **Microsoft SQL Server 2005 administrator's**. Microsoft Press, Redmond, WA, USA, 2006.

YIN, R. K. **Estudo de Caso: Planejamento e Métodos**. Porto Alegre: Bookman, 2005.