Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciência da Computação

UM ALGORITMO BASEADO EM GRAFO DE DERIVAÇÃO PARA REALIZAR FRAGMENTAÇÃO VERTICAL UNIDIMENSIONAL EM DATA WAREHOUSE

Autor: Artur Luis do Nascimento, aln@cin.ufpe.br
Orientador: Fernando da Fonseca de Souza, fdfd@cin.ufpe.br

Recife, 16 de março de 2005

Universidade Federal de Pernambuco
Centro de Informática

Artur Luis do Nascimento

UM ALGORITMO BASEADO EM GRAFO DE DERIVAÇÃO PARA REALIZAR FRAGMENTAÇÃO VERTICAL UNIDIMENSIONAL EM DATA WAREHOUSE

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Fernando da Fonseca de Souza

Recife, 16 de março de 2005

Aos meus pais, Hélio e Leninha, e aos meus irmãos, Halmos e Hugo, pelo apoio e incentivo que sempre me deram.

Agradecimentos

Agradeço a Deus, por sempre ter me dado forças nos momentos mais necessários. O meu orientador, Fernando da Fonseca de Souza, pelo incentivo e confiança. A Cristina Dutra de Aguiar Ciferri e a Diogo pela ajuda e dedicação. A Vanessa e aos meus amigos, Márcio, Marcelo, Tiago, Saulo e Jean, pelo apoio e todos que de uma forma ou de outra contribuíram pra a realização deste trabalho.

Artur Luis do Nascimento

Resumo

Este trabalho tem como objetivo resolver o problema da fragmentação vertical de um data warehouse através da escrita de um algoritmo baseado em grafos de derivação. Segundo Ciferri [Ciferri02], utiliza-se grafos de derivação por consistir de uma representação apropriada para o armazenamento dos dados do data warehouse em diferentes níveis de agregação. Para critério de fragmentação, por razões semânticas, serão considerados os atributos relativos a dados financeiros, que são de extrema importância para uma empresa.

Palavras-chave:

Data Warehouse, Fragmentação Vertical, Grafos de Derivação, Sistemas de Apoio à Decisão.

Abstract

This work aims to solve the problem of vertical fragmentation for a data warehouse through the writing of algorithm based on derivation graphs. As pointed out by Ciferri [Ciferri02], derivation graphs are used for consisting of an appropriate representation for storing data of a data warehouse in different aggregation levels. For fragmentation criterion, for semantic reasons, it will be considered attributes related financial data because they are of paramount importance for any company.

Keywords:

Data Warehouse, Vertical Fragmentation, Derivation Graph, Decision Support System.

Sumário

Lista de Figuras

Lista de Quadros

Lista de Abreviaturas

ADW – Ambiente de Data Warehouse

DM – Data Marts

DW – Data Warehouse

FV – Fragmentação Vertical

LAN – Local Area Network

OLAP – On-Line Analytical Processing

OLTP – On-Line Transaction Processing

SAD – Sistemas de Apoio à Decisão

SGBDM – Sistema de Gerenciamento de Banco de Dados Multidimensional

SIE – Sistemas de Informações Executivas

SSD – Sistemas de Suporte a Decisão

TI – Tecnologia de Informação

VPN – Virtual Private Network

WAN – Wide Area Network

WebD²W – Web Distributed Data Warehouse

1. Introdução

O desenvolvimento da tecnologia da informação, aliado à globalização e ao aumento da competitividade, nos mais diversos setores está facilitando a integração entre o mercado produtor e consumidor. A cada negócio realizado, ou seja, a todo o momento, uma grande quantidade de dados é gerada e armazenada, passando a ser um importante recurso da empresa.

Um dos principais problemas é que os processos de tomada de decisão na maioria das vezes envolvem uma grande quantidade de dados, caracterizando-se pela necessidade de determinar relacionamentos complexos entre variáveis. Sistema de Suporte à Decisão (SSD) é um sistema computacional integrado, interativo, consistindo de ferramentas analíticas e capacidades de gerenciamento da informação, projetado para auxiliar tomadores de decisão na solução de problemas relativamente grandes e não estruturados.

Data warehouse vem sendo consideravelmente utilizado nos últimos anos, com o objetivo de fornecer um ambiente adequado a estas análises. Segundo Inmon [Inmon97a], um data warehouse é um conjunto de dados baseados em assuntos, integrado, não-volátil, e variável em relação ao tempo, de apoio às decisões gerenciais.

A tecnologia de data warehouse vem sendo utilizada como uma ferramenta de suporte aos sistemas gerenciais de tomada de decisões, proporcionando descobertas de conhecimentos relevantes que podem ser de extrema importância para as organizações.

De acordo com Ciferri [Ciferri02], data warehouse representa uma única base de dados centralizada. Distribuir os dados armazenados nessa base de dados levando-se em consideração as características intrínsecas de aplicações de

data warehousing apresenta várias vantagens, porém introduz novos desafios a ambientes de data warehousing. Dentro deste contexto, foi proposto o sistema WebD²W que enfoca a distribuição dos dados do Data Warehouse.

Por ser um ambiente que contém informações importantes sobre toda a empresa o data warehouse necessita de algum tipo de segurança e somente poucos usuários têm autorização para ver dados em qualquer lugar do data warehouse [Singh97].

O processo de proteger um bem valioso da organização contra acesso não autorizado e fazer o dado disponível para qualquer um dentro da empresa pode ser bastante caro. Desta forma, seria interessante abordar uma fragmentação vertical do data warehouse focada tanto em dados financeiros quanto em confidenciais de uma empresa.

1.1. Objetivo

Este trabalho tem como principal objetivo desenvolver um algoritmo com a finalidade de resolver o problema da fragmentação vertical de um data warehouse em termos dos dados financeiros ou confidencias da empresa. Além de aumentar a segurança dos dados financeiros da empresa, que são de extrema importância para a organização, também há um ganho de desempenho no acesso a esses dados.

1.2. Estrutura do Trabalho

O trabalho se encontra divido em cinco capítulos. O primeiro capítulo tem como objetivo mostrar a idéia geral do trabalho, o objetivo a ser atingido e a estrutura de divisão dos capítulos. No segundo capítulo serão descritos os detalhes de um ambiente de data warehouse. O capítulo 3 tem como objetivo descrever a arquitetura do Sistema WebD²W e seus principais componentes. O

capítulo 4 descreve o algoritmo proposto para a fragmentação vertical dos dados do data warehouse de acordo com os dados financeiros ou confidenciais da empresa, baseado em grafos de derivação. O capítulo 5 apresenta as considerações finais e sugestões de trabalhos futuros.

2. Data Warehouse

Devido à rápida evolução da tecnologia de informação (TI) e a disseminação do uso de computadores ligados entre si, a maioria das organizações de médio e grande porte faz algum uso de sistemas informatizados para realizar seus processos mais importantes.

Uma enorme quantidade de dados relacionados aos negócios da empresa é formada com o passar do tempo, porém esses dados são desintegrados entre si. Estes dados armazenados servem como subsídio estratégico no seu estado original para a tomada de decisões.

Os sistemas tradicionais de informática não são desenvolvidos com a finalidade de gerar e armazenar informações estratégicas. Suas bases são formadas de dados cruciais à operação da organização. Em termos de decisão, os dados de certa forma são vazios e sem valor transparente para o processo gerencial das organizações. Estas decisões normalmente são tomadas baseadas na experiência dos administradores, quando pode, também, ser fundamentadas em fatos históricos que foram armazenados pelos diversos sistemas de informação utilizados pelas organizações.

Um DW é projetado para que os dados possam ser armazenados e acessados de maneira que não fiquem restritos a tabelas e linhas relacionais. Como o DW está separado dos bancos de dados operacionais, as consultas dos usuários não impactam nestes sistemas, que ficam resguardados de alterações indevidas ou de perdas de dados.

O DW contempla a base e os recursos necessários para um Sistema de Apoio à Decisão (SAD) e, principalmente, Sistemas de Informações Executivas (SIE) eficientes, fornecendo dados integrados e históricos que atendem desde a

alta direção, que necessita de informações mais resumidas, até as gerências de baixo nível, em que os dados detalhados ajudam a observar aspectos mais táticos da empresa. Nele, os executivos podem obter de modo imediato respostas para perguntas que normalmente não as possuem em seus sistemas operacionais e, com isso, tomar decisões com base em fatos, e não apenas em intuições ou especulações.

Desta forma, um DW provê um banco de dados especializado, que gerencia o fluxo de informações a partir de banco de dados corporativos e fontes de dados externas à organização.

2.1. Ambiente Operacional X Ambiente de Data Warehouse

O Ambiente de Data Warehouse (ADW) é fundamentalmente diferente do ambiente convencional de processamento de transações, ou seja, o ambiente operacional. As diferenças entre eles existem quanto aos sistemas, aos tipos de usuários, às tecnologias de suporte, ao volume de dados, ao histórico, à utilização da informação dentro do negócio e aos metadados.

O objetivo principal de um ambiente operacional é automatizar as principais tarefas de uma empresa, trabalhando com sistemas orientados a processos (On-Line Transaction Processing – OLTP). Enquanto que no ADW, o seu principal objetivo é disponibilizar sistemas com capacidade de acessar e fornecer informações que proporcionem apoio à tomada de decisão, corretamente e eficientemente. Estes sistemas são chamados de sistemas ou ferramentas OLAP (On-Line Analytical Processing).

O Quadro 1 apresenta as principais diferenças existentes entre o ambiente operacional e o ADW [Singh01].

Quadro 1 - OLTP versus data warehouse

Tópico ou Função	Operacional	Data Warehouse
Conteúdo dos dados	Valores atuais	Dados de arquivo, sumarizados e calculados
Organização dos dados	Aplicação por aplicação	Áreas de assunto ao longo da empresa
Estrutura dos dados; formato	Complexa;adequada para computação operacional	Simples;adequada para análise empresarial
Probabilidade de acesso	Alta	Moderada a baixa
Atualização dos dados	Campo a campo	Acessados e manipulados sem atualização direta
Uso	Processamento repetido e altamente estruturado	Processamento analítico e altamente desestruturado
Tempo de resposta	Fração de segundos	Segundos a minutos

2.2. Data Marts

Para Inmon [Inmon97a], um data mart pode ser definido como um SGBD multidimensional que fornece uma estrutura bastante flexível de acesso aos dados. Enquanto o data warehouse extrai, transforma e limpa os dados dos sistemas transacionais, mantendo-os integrados em quantidades massivas e em seu nível mais baixo, o data mart se serve destes dados, extraindo dados para um departamento ou uma área de negócio, oferecendo flexibilidade e controle ao usuário final, pois com o data mart é possível fatiar e agrupar dados de diversas maneiras.

Data mart (DM) é um subconjunto do data warehouse, direcionado a um departamento ou área de negócio. Para Kimball [Kimball98b], o conjunto de todos

os data marts da organização, construídos de forma incremental, compartilhando dimensões e fatos comuns, segundo um planejamento prévio, formam o data warehouse lógico da organização.

O DM é uma coleção de assuntos de uma determinada área, baseado nas necessidades de um departamento ou setor de uma empresa. Por exemplo, uma empresa terá um DM para o departamento financeiro e outro para o departamento de marketing.

Normalmente, todos os DM possuem hardware, software, dados e programas próprios. Esta característica de independência torna difícil o controle e coordenação dos dados localizados em DM diferentes. Devido a esta dificuldade, torna-se necessário a elaboração de um DM que trabalhe como um grande centralizador, ou de uma ferramenta que permita centralizar os dados distribuídos pelos diferentes DM. Segundo Inmon [Inmon98], os DM apresentam as seguintes características:

- São especificados para atender a uma área ou conjunto de áreas de interesse;
- Empregam normalmente um esquema estrela no projeto de banco de dados. Esta modelagem é elaborada com base nas exigências dos usuários finais;
- Contêm uma quantidade razoável de informações históricas, normalmente, menor que o volume histórico do DW;
- Apresentam uma granularidade, normalmente, maior que a do DW. Esta granularidade tem o propósito de atender às necessidades do usuário final; e
- Apresentam um armazenamento dos dados altamente indexado.

Existem dois tipos de DM, o DM dependente e o DM independente. O Quadro 2 apresenta as diferenças entre os dois tipos de DM [Inmon98].

Quadro 2 - Diferenças entre DM dependente e DM independente

DM Dependente	DM Independente
Fonte de dados é o data warehouse	Fonte de dados são os sistemas operativos do ambiente operativo.
Carga centralizada. Todos os DM são atualizados pela mesma fonte.	Carga descentralizada. Cada DM é atualizado de forma separada e exclusiva a partir do ambiente operativo.
Arquitetura de fácil crescimento	Difícil aproveitamento dos dados existentes. A arquitetura dificulta o crescimento.

Um dos principais problemas encontrados nos DM independentes está no fato de suas deficiências se manifestarem com a construção de múltiplos DM independentes, ou seja, após a implementação de alguns DM é possível perceber o problema.

2.3. Data Warehouse X Data Marts

Nos primórdios do ADW, muitos desenvolvedores de ferramentas de DM pregavam que os conceitos de DW e DM apresentavam o mesmo significado [Inmon98].

Este ponto de vista errado propiciou que várias empresas desenvolvessem o ADW a partir dos DM, sem se preocuparem muito com o desenvolvimento de um DW e com o controle dos dados. As observações destas experiências demonstraram que a ausência de um DW implica, dentre outras [Inmon98]:

- Redundância de grande volume de dados detalhados e históricos de um DM para outro;

- Resultados incompatíveis e irreconciliáveis de um DM para o outro; e
- Uma interface intratável entre os DM e o ambiente operativo.

Atualmente, muitos desenvolvedores de software apresentam o DW como uma coleção de DM integrados. Porém, a integração de múltiplos DM não consiste em uma simples tarefa. Os DM são desenvolvidos com a finalidade de atender a usuários específicos, como, por exemplo, um departamento ou setor, sem necessitar da integração com outros DM.

O Quadro 3 apresenta as principais diferenças entre Data Warehouse e Data Marts [Inmon98].

Quadro 3 - Diferenças entre o Data Warehouse e o Data Mart

Data Warehouse	Data Mart
Corporativo	Departamental
Granularidade em baixo nível. Dados bem detalhados	Granularidade em alto nível
Estrutura normalizada (com tratamento)	Emprega o esquema estrela como estrutura de dados
Grande volume de histórico de dados	Não armazena grande volume de dados históricos
Emprega tecnologia orientada ao armazenamento de grandes volumes de dados	Emprega tecnologia multidimensional excelente para acesso e análise
Modelagem de dados com o propósito de atender à corporação	Modelagem de dados com o objetivo de atender a um usuário final
Levemente indexado	Altamente indexado

2.4. Características do data warehouse

Data warehousing é um conjunto de ferramentas e técnicas de projeto, que quando aplicadas às necessidades específicas dos usuários e aos bancos de dados específicos permitirá que planejem e construam um data warehouse [Kimball00].

De acordo com Inmon [Inmon97a], para dar suporte ao processo gerencial de tomada de decisão, um data warehouse deve possuir as seguintes características:

- Orientado por assuntos - consiste em uma das características mais importantes do data warehouse, pois toda modelagem do data warehouse é voltada em torno dos principais assuntos da empresa, como: vendas, serviços, entre outros. Enquanto todos os sistemas transacionais estão voltados para processos e aplicações específicas, os DW objetivam assuntos, ou seja, um conjunto de informações relacionadas com uma determinada área estratégica de uma empresa;

- Integração - é através dessa característica que é definida uma representação única para os dados de todos os sistemas que formarão a base de dados do data warehouse. Por isso, uma boa parte do trabalho na construção de um data warehouse está na análise dos sistemas transacionais e dos dados que eles contêm. Esses dados geralmente encontram-se armazenados em vários padrões de codificação. Isso se deve aos inúmeros sistemas existentes nas empresas, e ao fato deles terem sido projetados por diferentes analistas. Isso quer dizer que os mesmos dados podem estar em formatos diferentes. Os dados referentes aos gêneros masculino e feminino podem estar representados de formas diferentes, como: M (masculino) e F (feminino), ou H (homem para masculino) e M

(mulher para feminino). As mesmas informações estão em formatos diferentes, e isso em um DW, não pode acontecer. Portanto, é por isso que deverá existir uma integração de dados, definindo-se uma maneira uniforme de se armazenar os mesmos, e isso gera bastante trabalho;

- Variável no tempo - é outra característica de um data warehouse. Os data warehouses são variáveis em relação ao tempo, ou seja, eles mantêm o histórico dos dados durante um período de tempo muito superior ao dos sistemas transacionais. Em um sistema transacional, a finalidade é de fornecer as informações no momento exato. Já no data warehouse, o principal objetivo é analisar o comportamento das mesmas durante um período de tempo maior. Assim, os gerentes tomam as decisões em cima de fatos e não de intuições; e

- Não volatilidade - no data warehouse existem somente duas operações: a carga inicial e as consultas aos dados. Isso porque o carregamento e o tratamento dos dados são diferentes em relação aos sistemas transacionais. Por exemplo, em um sistema de contabilidade pode-se fazer alterações nos registros. Já no DW, o que acontece é somente a leitura dos dados na origem e a gravação deles.

Antes dos dados serem inseridos no data warehouse, eles sempre passam por filtros. Com isso muitos deles jamais saem do ambiente transacional. A maior parte dos dados é física e radicalmente alterada quando passa a fazer parte do DW. De acordo com Inmon [Inmon99], dificilmente ocorre a redundância de dados entre os dois ambientes, resultando em menos de 1 por cento de duplicações.

A localização dos dados pode estar fisicamente armazenada de três formas:

- Centralizado - banco de dados num DW integrado, com a finalidade de maximizar o poder de processamento e agilizar a busca dos dados. Esse tipo de armazenamento é bastante utilizada. Porém, há o inconveniente do investimento em hardware para comportar a base de dados muito volumosa, e o poderio de processamento elevado para atender satisfatoriamente às consultas simultâneas de muitos usuários;

- Distribuído - vários DM, armazenados por áreas de interesse. Por exemplo, os dados financeiros num servidor, dados de marketing noutro e dados da contabilidade num terceiro lugar. Essa pode ser uma saída interessante para quem precisa de bastante desempenho, pois isso não sobrecarrega um único servidor, e as consultas poderão ser sempre atendidas em tempo satisfatório; e

- Níveis de detalhes - as unidades de dados são mantidas no DW. Armazenam-se dados altamente resumidos em um servidor, dados resumidos noutro nível de detalhe intermediário no segundo servidor e os dados mais detalhados (atômicos), num terceiro servidor. Os servidores da primeira camada podem ser otimizados para dar suporte a um grande número de acessos e um baixo volume de dados, enquanto alguns servidores nas outras camadas podem ser adequados para processar grandes volumes de dados, mas prevendo um baixo número de acessos. Cada nível de detalhe possui um horizonte de tempo definido para a permanência dos dados. Então, o fato dos dados serem transportados para níveis mais elevados não implica na exclusão do nível anterior. O processo de envelhecimento ocorre quando este limite é ultrapassado, e portanto os dados podem ser transferidos para meios de armazenamentos alternativos ou passar de dados detalhados atuais para dados detalhados antigos.

A credibilidade dos dados tem um alto grau de importância para o sucesso de qualquer projeto. Qualquer discordância pode causar problemas graves quando

se deseja extrair dados para suportar decisões estratégicas para o negócio das empresas. Dados de má qualidade resultam em um suporte à decisão de baixo nível com alto risco para o negócio da empresa.

2.5. Granularidade

Consiste no nível de detalhe dos dados existentes em um data warehouse. Quanto maior for o nível de detalhes, menor será o nível de granularidade. O nível de granularidade afeta diretamente o volume de dados armazenados no DW, e ao mesmo tempo o tipo de consulta que pode ser respondida.

O espaço em um disco e o número de índices necessários são menores quando se tem um nível de granularidade muito alto, porém há uma correspondente diminuição da possibilidade de utilização dos dados para atender a consultas detalhadas.

A Figura 1 exemplifica o conceito de granularidade, utilizando-se os dados históricos das vendas de um produto. Cada uma das vendas ocorridas para este produto é caracterizada por um nível de granularidade muito baixo, entretanto os somatórios das vendas ocorridas por mês podem ser caracterizados por um nível muito alto de granularidade.

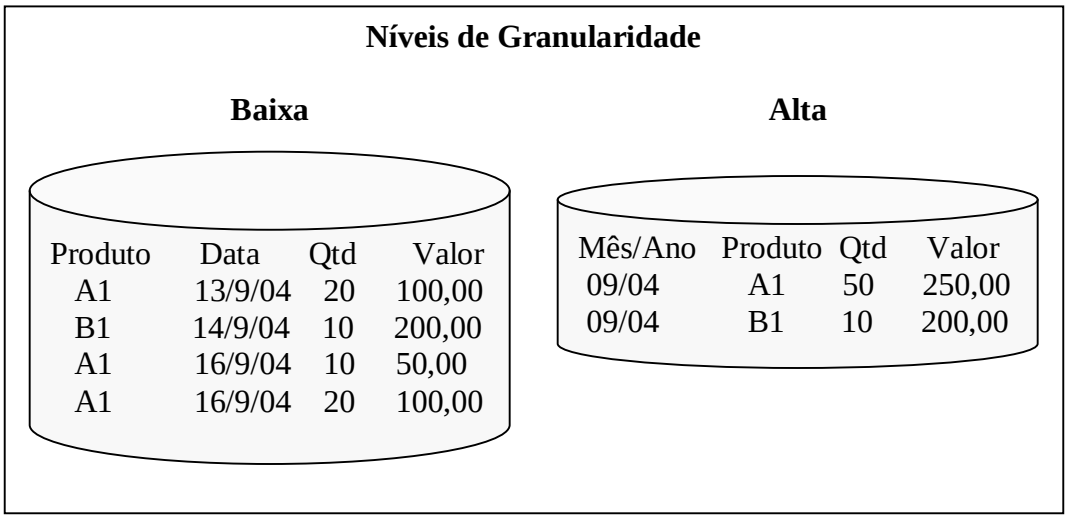

Figura 1 - Níveis de granularidade

É possível responder a qualquer consulta com o nível de granularidade muito baixo, mas exige uma maior quantidade de recursos computacionais. Como em um data warehouse, um evento isolado é raramente observado, é mais provável que ocorra a utilização da visão de um conjunto de dados.

Um nível intermediário na estrutura do data warehouse é formado por dados levemente resumidos. Na passagem para este nível, os dados sofrem modificações. Por exemplo, se as informações nos dados detalhados atuais são armazenadas por dia, nos dados levemente resumidos estas informações podem estar armazenadas por semanas. Neste nível, o horizonte de tempo de armazenamento normalmente fica em cinco anos e após este tempo, os dados sofrem um processo de envelhecimento e podem passar para um meio de armazenamento alternativo.

Os dados altamente resumidos são compactos e devem ser de fácil acesso, pois fornecem informações estatísticas valiosas para os SIE, enquanto que nos níveis anteriores ficam as informações destinadas aos SAD, que trabalham com dados mais analíticos procurando analisar as informações de forma mais ampla.

Um dos aspectos mais críticos no planejamento de um DW é conseguir o equilíbrio no nível de granularidade, pois na maior parte do tempo há uma grande demanda por eficiência no armazenamento e no acesso aos dados, bem como pela possibilidade de analisar dados em maior nível de detalhes. Quando uma organização possui grandes quantidades de dados no DW, faz sentido pensar em dois ou mais níveis de granularidade, na parte detalhada dos dados. Na realidade, a necessidade de existência de mais de um nível de granularidade é tão grande, que a opção do projeto que consiste em duplos níveis de granularidade deveria ser o padrão para quase todas as empresas.

O chamado nível duplo de granularidade se enquadra nos requisitos da maioria das empresas. Na primeira camada de dados ficam os dados que fluem do armazenamento operacional e são resumidos na forma de campos apropriados para a utilização de analistas e gerentes. Na segunda camada, ou nível de dados históricos, ficam todos os detalhes vindos do ambiente operacional. Como há uma verdadeira montanha de dados neste nível, faz sentido armazenar os dados em um meio alternativo como fitas magnéticas.

Com a criação de dois níveis de granularidade no nível detalhado do DW, é possível atender a todos os tipos de consultas, pois a maior parte do processamento analítico dirige-se aos dados levemente resumidos que são compactos e de fácil acesso. E para ocasiões em que um maior nível de detalhe deve ser investigado, existe o nível de dados históricos. O acesso aos dados do nível histórico de granularidade é caro, incômodo e complexo, mas caso haja necessidade de alcançar esse nível de detalhe, ele estará disponível.

2.6. Arquitetura de um data warehouse

De acordo com Singh [Singh01], arquitetura é um conjunto de regras que regulamentam uma estrutura para um produto ou projeto de um sistema. Os fundamentos da arquitetura de DW são muito mais importantes do que as

ferramentas específicas para implementação da mesma. O principal fundamento de qualquer DW é ter flexibilidade suficiente para que a organização de acordo com as suas necessidades possa modificar e analisar os seus dados. A arquitetura de DW deve ser custo-eficiente, adaptável e de fácil implementação. A Figura 2 mostra quais componentes de uma arquitetura devem ser considerados antes da implementação de um data warehouse.

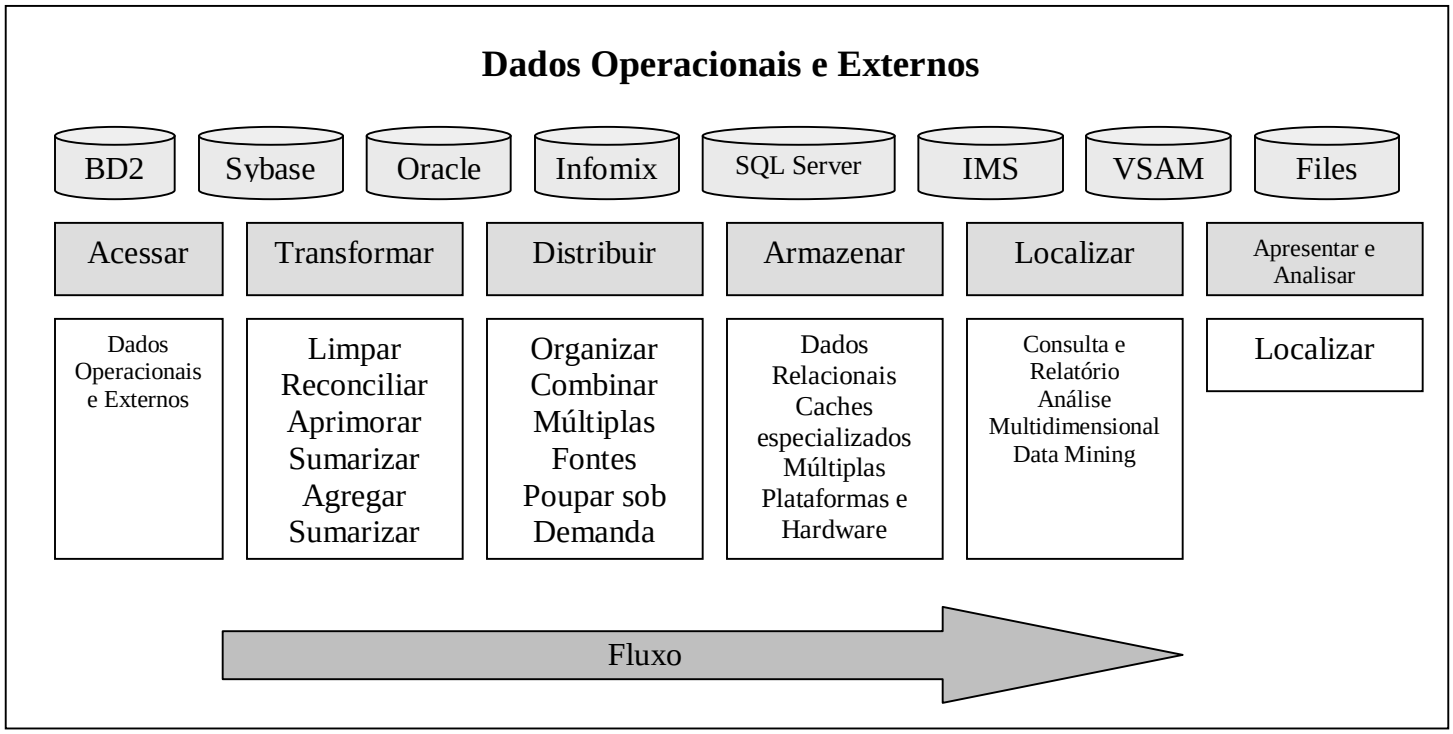

Figura 2 - Componentes de uma arquitetura que devem ser considerados antes da implementação do data warehouse [Singh01]

Kimball [Kimball98a] destaca a importância do plano de arquitetura na construção do projeto de data warehouse como ferramenta de comunicação, planejamento e flexibilidade, que facilita o aprendizado e aumenta a produtividade.

O projeto de arquitetura do data warehouse deve ser voltado para satisfazer a necessidade de informação da organização, não se restringindo por nenhuma tecnologia. Uma arquitetura física de cliente/servidor possibilita separar a aquisição de dados, os requerimentos de processamento, bem como o acesso a dados e as funções de processamento de manipulação de dados [Inmon99].

2.6.1.Arquitetura genérica de um data warehouse

A escolha da arquitetura é uma decisão gerencial do projeto e está relacionada com a infra-estrutura disponível e com o ambiente de negócios. Uma outra decisão importante é a abordagem de implementação, que provoca impactos quanto ao sucesso de um data warehouse.

Deve-se realizar uma análise detalhada em relação à escolha da arquitetura e à abordagem de implementação, analisando quanto tempo é necessário para a execução do projeto, o retorno do investimento, as vantagens na utilização das informações, a satisfação dos usuários executivos e finalmente, quais recursos serão necessários para a implementação de uma arquitetura.

A Figura 3 representa uma arquitetura típica de um ambiente de data warehousing [Ciferri02].

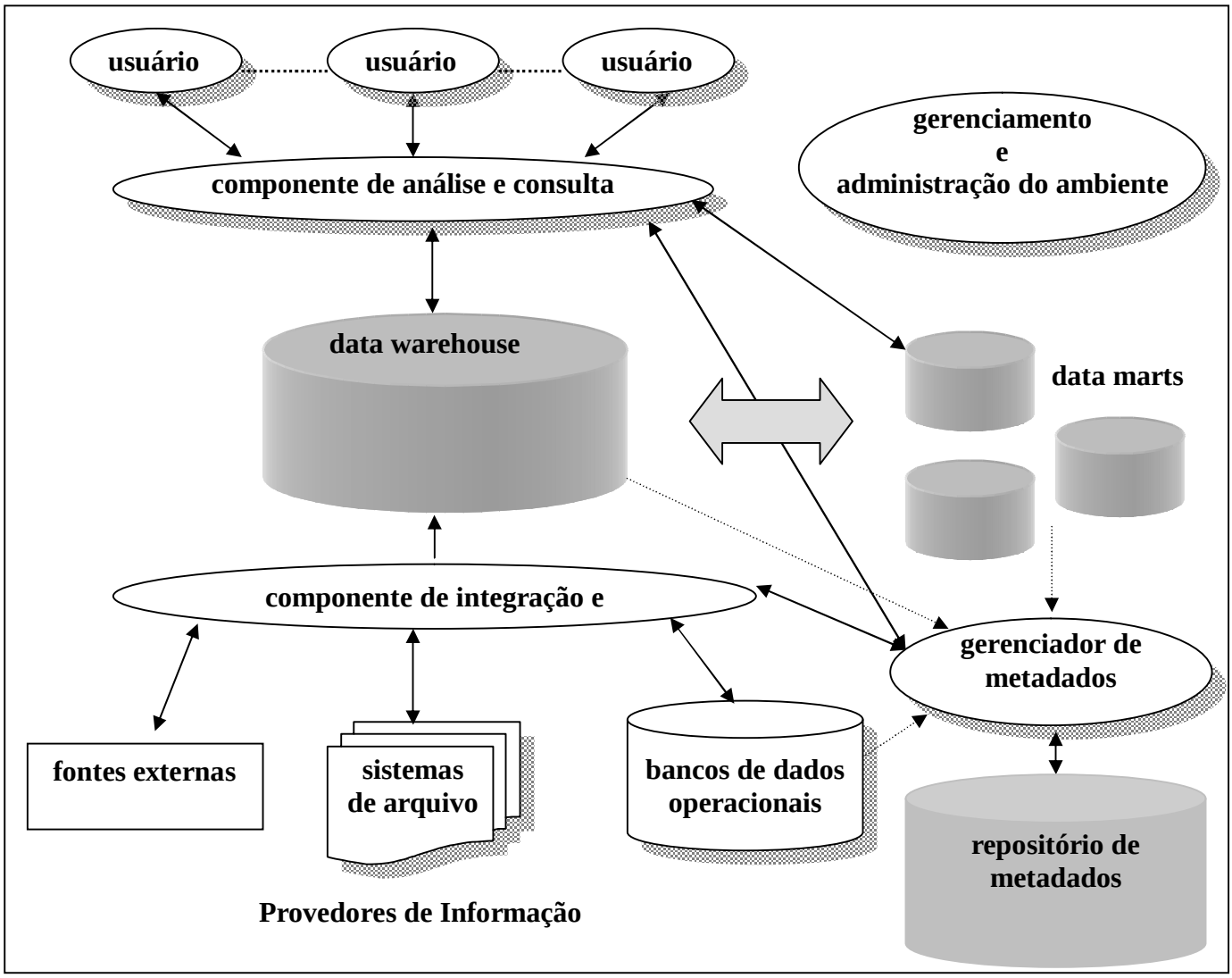

Figura 3 - Arquitetura típica de um ambiente de data warehousing [Ciferri02]

Os dados operacionais estão localizados nos provedores de informação, os quais contêm dados em diferentes modelos e formatos. O componente de integração e manutenção é responsável pela extração, tradução, filtragem, integração e armazenamento dos dados no data warehouse, além disso, ele também deve manter o data warehouse consistente.

Além do data warehouse principal, podem existir porções de dados do mesmo, ou seja, um conjunto de data marts que representam os fragmentos ou réplicas do data warehouse principal. É através do componente de análise e

consulta que as consultas dos usuários de SSD são submetidas e redirecionadas ao data warehouse principal ou aos data marts.

O gerenciador de metadados controla o repositório de metadados, que contém todas as informações estruturais e semânticas dos provedores de informação e do data warehouse.

Além desses componentes apresentados, a arquitetura também disponibiliza ferramentas para o gerenciamento e administração do ambiente. É através dela que o sistema é controlado, executando tarefas de extrema importância como o gerenciamento dos metadados e de segurança, testes de qualidade dos dados, backup, auditoria e a descrição da utilização do data warehouse para o controle do tempo de resposta e da utilização dos recursos.

2.7. Estratégias para a implementação de data warehouse

Basicamente existem três estratégias na fase de planejamento arquitetônico para o desenvolvimento de uma data warehouse, a estratégia top-down, a bottom-up e a intermediária. A seleção da estratégia é de fundamental importância na escolha da tecnologia adequada para o desenvolvimento do ambiente de data warehouse.

2.7.1.Estratégia Top-Down

Inicialmente, cria-se um data warehouse. Depois é realizada a fragmentação do mesmo em várias partes, gerando diversos bancos de dados orientados por assunto, ou seja, pequenos bancos de dados departamentais que consistem os data marts.

O data warehouse faz a integração dos dados utilizando todos os recursos existentes numa organização, como também todos os dados necessários para o suporte à decisão.

Esta estratégia apresenta várias vantagens importantes no ambiente de data warehouse, como as mostradas abaixo:

- Facilidade de manutenção devido a todos os data marts serem originados a partir do data warehouse;
- Possibilidade de extração de níveis menores de informações a partir do data warehouse, devido ao mesmo concentrar todos os negócios da empresa; e
- Necessidade de apenas um único conjunto de aplicações para realizar a extração, limpeza e integração dos dados.

Porém, esta estratégia possui algumas desvantagens, como:

- Longo período de implementação;
- Taxa de risco elevada; e
- Possível frustração das expectativas dos usuários devido ao longo projeto, possibilitando o abandono do projeto.

2.7.2.Estratégia Botton-Up

Nessa estratégia, o data warehouse é construído a partir da integração dos data marts independentes. Várias organizações preferem inicialmente criar um banco de dados em uma determinada área ou departamento, ou seja, os data marts. Assim, é possível manter todos os bancos de dados departamentais e integrá-los para desenvolver o data warehouse. Os custos de desenvolvimento de um data warehouse que utiliza esta estratégia são bem menores.

Esta estratégia possui as seguintes vantagens:

- Rápido desenvolvimento;
- Alto nível de confiança;
- Rápida apresentação dos resultados; e
- Permite que os principais negócios da empresa sejam analisados inicialmente.

Apesar das vantagens apresentadas acima, esta estratégia também apresenta algumas peculiaridades. Entre elas, pode-se destacar:

- Necessidade de um maior controle do negócio da empresa;
- Maior quantidade de trabalho a ser realizado pelos desenvolvedores para extrair e combinar as fontes individuais;
- Deve-se organizar esforços e recursos de várias equipes; e
- Data marts desvinculados do ambiente podem ser produzidos.

2.7.3.Estratégia Intermediária

Esta estratégia tem o objetivo de integrar a estratégia Top-Down com a Bottom-Up. Nesta abordagem, efetua-se a modelagem de dados do data warehouse, sendo o passo seguinte a implementação de partes desse modelo. Essas partes são escolhidas por área de interesse e constituem os data marts. Cada DM gerado a partir do modelo de dados do DW é integrado no modelo físico do data warehouse. A principal vantagem desta estratégia é a garantia de consistência dos dados. Essa garantia é obtida em virtude do modelo de dados para o DW ser único, possibilitando realizar o mapeamento e o controle dos dados [Hackney98].

2.8. Metadados

Singh [Singh97] fala que os metadados são o principal componente do Data Warehouse. A definição mais comum que se encontra na literatura sobre metadados é que eles representam "dados sobre dados". De uma forma um pouco mais completa pode-se dizer que o metadado é a "descrição do dado, do ambiente onde ele reside, de como ele é manipulado e para onde é distribuído".

Metadados são uma abstração dos dados, e permite que os dados armazenados em vários formatos tenham um determinado significado. Eles descrevem ou qualificam outro dado, incorporando, a este, um significado. Sem metadado, a informação se restringe a um conjunto de dados sem significado.

Com a utilização de metadados é possível avaliar o impacto das mudanças nos sistemas transacionais. Consequentemente a manutenção desses sistemas se torna menos complexa. Assim, os metadados assistem o processo de construção e manutenção do Data Warehouse.

O processo de construção de um DW é constituído de várias tarefas, onde a extração de dados é uma delas. Assim, a integração de todos os dados exige o conhecimento dos seus significados, estruturas, locais de armazenamento e sistemas que os mantêm atualizados. Os metadados devem possuir todo esse conhecimento.

A carência de metadados integrados, competentes em descrever os dados totalmente, dificulta ainda mais a integração e o compartilhamento dos dados nas empresas.

Os metadados assumem um papel de extrema importância no processo de transformação dos dados, pois através deles, serão armazenadas as lógicas das

transformações e os mapeamentos entre os dados dos sistemas transacionais e aqueles mantidos no DW.

Dentre as funções dos metadados, pode-se destacar as seguintes:

- Aumentar a produtividade do DW, pois uma vez definido, um atributo poderá ser reutilizado [Inmon97b];
- Permitir ao usuário final e aos desenvolvedores, uma "navegação" pelos dados. Isto torna possível descobrir a origem de uma informação e as regras empregadas para integrá-la no ADW [Melo97];
- Permitir ao DW e aos DM, apresentarem atributos com termos empregados pelos analistas de suporte e apoio à decisão;
- Gerenciar o mapeamento entre o ambiente operativo, o DW e os DM. A gerência de mapeamento de dados engloba as conversões, filtragens, alterações estruturais e qualquer outra informação necessária ao rigoroso acompanhamento das transformações; e
- Manter o acompanhamento das alterações estruturais dos dados ao longo dos anos.

Para garantir o cumprimento dessas funções, os metadados devem manter informações sobre [Inmon97a, Melo97]:

- Estrutura de dados, segundo a visão do programador;
- Estrutura de dados, segundo a visão do usuário final;
- Fonte de dados que alimentam o DW;
- Transformações sofridas pelos dados no momento de sua migração para o DW;
- Transformações sofridas pelos dados no momento de sua migração para o DM;
- Modelo de dados;
- Relacionamento entre o modelo de dados, o DW e os DM; e

- Histórico de extrações.

2.8.1.Metadados técnicos X Metadados de negócios

Metadados técnicos proporcionam aos desenvolvedores e usuários técnicos de sistemas de suporte à decisão, a confiança de que o dados estão corretos. Eles são críticos para a manutenção e o crescimento contínuos de um data warehouse. Sem metadados técnicos, a tarefa de analisar e programar mudanças em um SSD é significativamente mais difícil e consome mais tempo.

Os metadados de negócios são as ligações entre o DW e os usuários de negócios. Esses dados fornecem uma espécie de mapa aos usuários para que eles possam acessar os dados, tanto no DW como nos DM. Os usuários de negócios são basicamente executivos ou analistas de negócios e com isso tendem a ser menos técnicos. Portanto, eles precisam ter o SSD definido para eles em termos de negócios. Os metadados de negócios mostram em termos de negócios, que relatórios, consultas, dados estão no data warehouse, localização dos dados, confiabilidade dos dados, contexto dos dados, regras de transformação que foram aplicadas e quais as origens desses dados.

Os projetos de data warehouse e data mart necessitam ter um repositório de metadados como parte de seus objetivos principais, desde o início do projeto. Esse repositório precisa ser construído com uma tecnologia confiável e considerando os usuários de negócios. Além disso, os metadados necessitam de mecanismos para os usuários não-técnicos poderem navegar e acessar as informações no repositório.

O repositório de metadados ajuda significativamente o SSD a tornar as informações no data warehouse e nos data marts mais visíveis, inteligíveis e acessíveis aos usuários. Em resumo, um repositório pode viabilizar ou não um DW.

A Figura 4 mostra o esquema básico de um repositório de metadados com a distinção entre os metadados técnicos e os metadados de negócios.

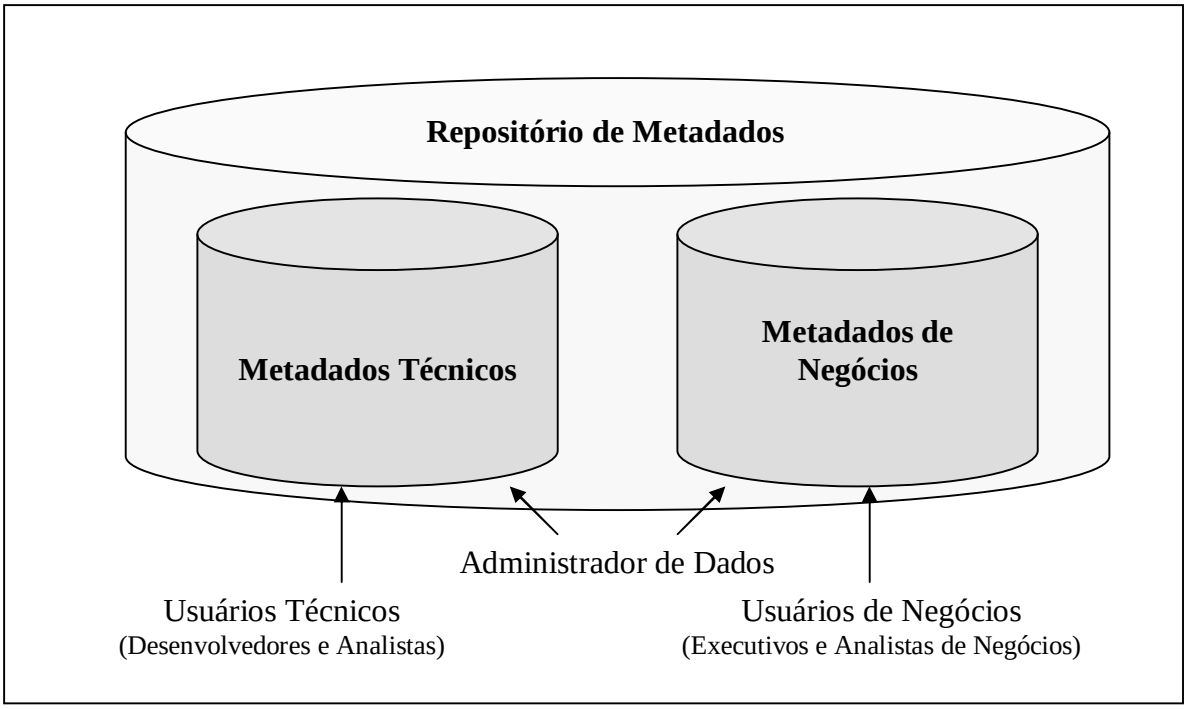

Figura 4 - Repositório de Metadados

2.9. Data warehousing virtual

No ambiente de data warehousing virtual, não existe fisicamente um data warehouse, diferentemente de um ambiente de data warehousing convencional. O data warehouse virtual utiliza o conceito de metadados com regras de negócios. A partir dos metadados são acessados os dados diretamente das bases operacionais. Isto provoca uma redução nos custos, como também uma diminuição no tempo de implementação, ou seja, são mais simples de implementar.

É através de ferramentas de análise e consulta, que os dados operacionais são pesquisados nos provedores de informação e retornados para os usuários como se os dados tivessem sido obtidos de um data warehouse consolidado.

A Figura 5 apresenta a arquitetura básica de um data warehousing virtual, onde não estão presentes alguns componentes importantes do data warehousing convencional.

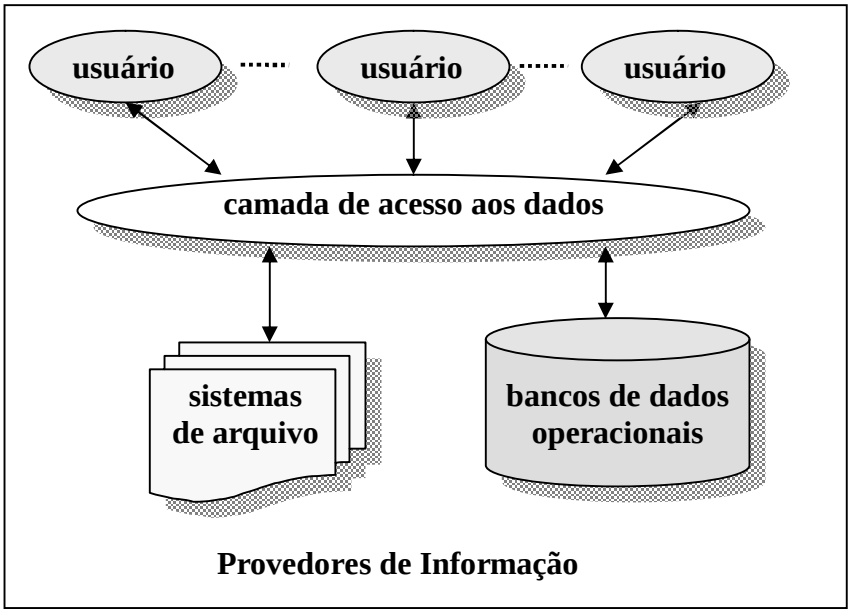

Figura 5 - Arquitetura básica de uma ambiente de data warehousing virtual

2.10. Data warehousing virtual X Data warehouse convencional

Segundo Ciferri [Ciferri02], ambientes de data warehousing virtuais são muito menos robustos do que ambiente de data warehousing convencionais. Assim, os data warehousing virtuais apresentam algumas desvantagens e limitações em relação aos data warehousing convencionais.

Os ambientes de data warehousing virtuais no geral oferecem funcionalidades analíticas mais simples e limitadas comparando-se com a dos ambientes de data warehousing convencionais. Além disso, os dados obtidos por estes ambientes são altamente voláteis e não orientados a assunto, prejudicando as análises comparativas complexas e de tendência.

No processo de carregamento dos dados nos data warehousing convencionais ocorre uma limpeza dos dados, resultando em dados sobre o negócio com uma maior qualidade. Como nos data warehousing virtuais, os dados são obtidos diretamente dos provedores de informação, a qualidade dos dados fica dependente da qualidade dos dados dos provedores de informação, que podem conter dados incorretos. Como resultado, os usuários podem receber respostas indesejadas.

O processo de integração dos dados operacionais em ambientes de data warehousing convencionais é realizado pelo componente de integração e manutenção. Como não existe este componente na arquitetura dos ambientes de data warehousing virtuais, o processo de integração dos dados deve ser manipulado pelas ferramentas de análise e consulta. Cada ferramenta pode solucionar os conflitos estruturais existentes entre os provedores de informação, porém, isto introduz incompatibilidade e redundância de esforços.

Nos ambientes de data warehousing convencionais, o data warehouse contém tanto dados detalhados, como também dados resumidos, ou seja, agregados. Porém, nos ambientes de data warehousing virtuais, só existem dados detalhados, que são acessados diretamente a partir dos provedores de informação. Os dados operacionais não são estruturados com a finalidade de se obter um bom desempenho no processamento de consultas OLAP, provocando um aumento nos custos de entrada/saída e de processamento. Assim, o tempo de resposta aos usuários de SSD são maiores nos ambientes de data warehousing virtuais.

Como nos ambientes de data warehousing virtuais não ocorre a separação entre os ambientes operacional e informacional da empresa, ao contrário do ambiente de data warehousing convencional, o desempenho do ambiente operacional também é comprometido. Isto ocorre porque as consultas OLAP

disputam pelos mesmos recursos com as transações OLTP, resultando em maiores tempos de resposta aos usuários.

2.11. Modelos de dados

A modelagem dimensional é uma técnica de modelagem de dados voltada especialmente para a implementação de um modelo de dados que permita a visualização de dados de forma intuitiva e com altos índices de performance na extração de dados [Kimball98b]. Este modelo proporciona uma representação ideal do banco de dados consistente, determinando como o usuário visualiza e navega pelo data warehouse. O modelo dimensional é baseado em três elementos:

- Fatos - consiste em uma coleção de itens de dados composta de dados de medida e dados de contexto. Dados de medida são as medições numéricas do negócio e os dados de contexto são chaves estrangeiras apontando para cada uma das dimensões. Cada fato pode representar uma determinada transação ou evento do negócio ocorrido em um determinado contexto obtido pela intersecção das dimensões;
- Dimensões - se referem ao contexto em que um determinado fato ocorreu, tais como períodos de tempo, produtos, clientes e fornecedores. Estes elementos descrevem o contexto de um fato específico e classifica as medições ativas de uma organização; e
- Medidas - são atributos que quantificam um determinado fato, representando a performance de um indicador em relação às dimensões que participam do fato. O contexto de uma medida é determinado em relação às dimensões que participam do fato.

Um cubo pode representar um modelo dimensional para a visualização dos dados, permitindo que cada dimensão do cubo represente o contexto de um fato específico. A representação das medidas de um fato determinado é realizada

através da intersecção entre as dimensões do cubo. Um cubo possui três dimensões, porém, no modelo dimensional a metáfora do cubo pode possuir várias dimensões, de acordo com a necessidade de se representar um determinado fato.

2.12. Tendências do mercado de data warehouse

O mercado de data warehouse é um dos mais ativos no momento, considerando-se que a maior parte dos fornecedores de sistemas de gerenciamento de banco de dados e ferramentas estão lançando produtos novos e mais sofisticados em quantidade considerável [Kimbll00].

É nestes produtos de software que se espera que ocorram os grandes avanços na área, especialmente com relação à melhoria de desempenho e qualidade das ferramentas para o usuário final. Kimball [Kimball96] lista alguns dos pontos que deverão melhorar significativamente nos próximos anos:

- Otimização das estratégias de execução para consultas de junção em esquema estrela;
- Indexação de tabelas de dimensão, em especial tabelas de muitos milhões de linhas;
- Acesso e indexação da chave composta de grandes tabelas de fatos;
- Extensão de SQL para processar consultas do estilo OLAP;
- Suporte a processamento paralelo;
- Ferramentas para projeto de bancos de dados multidimensionais;
- Ferramentas para extração e administração; e
- Ferramentas de consulta do tipo data mining para o usuário final.

O desenvolvimento de servidores paralelos de BD poderá viabilizar o suporte a data warehouses cada vez maiores, permitindo um processamento mais rápido de suas dispendiosas consultas. O processamento paralelo acelera

dramaticamente o processamento distribuindo a execução de uma tarefa por múltiplos processadores. Avanços no sentido de tratar também dados multimídia virão ampliar a fronteira de suas aplicações.

O data warehouse deverá também ser viabilizado no ambiente da Web. O baixo custo por usuário e o acesso facilitado da Web já atingiram as mais diversas áreas de aplicação, podendo já se ver iniciativas por parte dos fornecedores no sentido de viabilizar o acesso aos data warehouses via Web.

2.13. Segurança dos dados no data warehouse

Ninguém em uma organização é mais bem equipado para qualificar usuários e conceder direitos de acesso ao DW do que a própria gerência do data warehouse. Logo ela deve gerenciar ativamente a segurança do DW, dominando o assunto para que possa orientar especialistas em segurança [Kimball98b].

Uma das ironias do trabalho da gerência de DW é a tensão entre o dever de publicação e o dever de proteção. A gerência de DW tem sido confiada com os dados da organização, mas pode ser responsabilizada se eles forem perdidos ou roubados [Kimball00].

Para garantir a segurança do DW é necessário utilizar ferramentas de ambiente de rede que dão suporte para segurança, como: criptografia, firewalls, servidores de diretório, entre outros. Além dessas ferramentas, deve-se estabelecer uma política de controle de acesso ao DW, restringindo o acesso aos dados de acordo com o nível de permissão do usuário.

Segundo Kimball [Kimball00], para aumentar o nível de segurança dos dados, devem ser seguidos alguns elementos, como:

- Consciência;

- Suporte a executivos;
- Policiamento;
- Vigilância;
- Desconfiança; e
- Renovação.

De acordo com Kimball e Merz [Kimball00], a estrutura de segurança do data warehouse deve se basear em quatro elementos:

- Dois fatores para autenticação;
- Uma conexão segura;
- Definição forte dos papéis do usuário; e
- Acesso a todos os objetos do Warehouse controlados por papéis.

Os gerentes de DW devem proteger as informações da organização, como também publicá-las. Eles devem tomar medidas para acabar com os problemas pré-existentes da organização e devem tomar decisões estratégicas, planejando e atualizando as medidas no decorrer do tempo.

Para aumentar a segurança no acesso aos dados, uma senha de acesso deve ser fornecida ao usuário, além de um cartão magnético de identificação, aperfeiçoando assim a autenticação do usuário. Com isso, são definidos os dois fatores para a autenticação do usuário.

É de fundamental importância que a conexão seja segura, não permitindo que nenhum aplicativo consiga monitorar os dados que trafegam pela rede. Para isto, as comunicações nas redes LAN (Local Area Network) devem ser criptografadas e para redes WAN (Wide Area Network) deve ser utilizado técnicas de VPN (Virtual Private Network).

Os usuários devem ser organizados em grupos de interesse, fornecendo assim direitos de acesso de informações no DW para cada classe de usuário, onde os papéis dos usuários são definidos para cada grupo.

2.14. Conclusão

Diversos são os fatores que vêm influenciando a absorção do DW pelas empresas. Um dos pontos considerados fundamentais é a busca da vantagem competitiva das empresas. Esta vantagem, atualmente, reside na capacidade de tomar decisões estratégicas e táticas de forma ágil, com base nas informações disponíveis para os tomadores de decisões das empresas.

Desta forma, foram apresentados neste capítulo conceitos relacionados ao DW, como a arquitetura típica de um DW com os seus componentes e suas respectivas funcionalidades. Além disso, foi mostrado as principais estratégias de implementação de um DW, as diferenças entre o DW e DM, entre outros.

O capítulo 3 apresenta a arquitetura do sistema WebD²W e seus principais componentes, onde o principal objetivo é aumentar a disponibilidade dos dados no ambiente distribuído.

3. Sistema WebD²W

O sistema WebD²W (Web Distributed Data Warehouse) foi projetado com uma arquitetura cliente-servidor utilizando a Web como infra-estrutura para facilitar a distribuição dos dados, como também o acesso aos mesmos.

Este sistema visa alcançar vários objetivos, que juntos proporcionarão uma grande melhoria para os usuários de sistemas de suporte à decisão. Segundo Ciferri [Ciferri02], um dos objetivos principais do sistema consiste em aumentar a disponibilidade dos dados e do acesso aos mesmos. Isto é possível através do uso de metodologias e algoritmos para a distribuição dos dados do data warehouse para diversos sites, aumentando o desempenho das consultas dos usuários de SSD. Com isso, é possível aumentar a quantidade de consultas aos dados, devido à utilização da Web como infra-estrutura, pois os dados podem estar replicados em diversos sites do ambiente de data warehouse.

Caso ocorra alguma falha em um site específico, outro site que contém a réplica dos dados pode fornecê-los quando requisitados pelos usuários, satisfazendo as necessidades desse usuário. Porém, isto introduz um problema, que consiste em manter a consistência dos dados, ou seja, os dados distribuídos devem estar sempre atualizados. Para isso, o sistema deve atualizar todas as réplicas dos dados caso ocorra alguma atualização em algum site.

Além disso, o sistema também deve garantir as transparências de fragmentação, de replicação e de localização. Isto significa que os usuários de SSD podem realizar consultas OLAP no ambiente de data warehousing distribuído como se o ambiente fosse constituído de apenas um data warehouse centralizado. Devido à distribuição dos dados em diversos sites e à possibilidade de acesso local aos sites, o sistema aumenta o desempenho no processamento de consultas OLAP.

Como resultado, o sistema provê suporte a um número elevado de usuários através da utilização da Web como meio de acesso aos dados.

3.1. Arquitetura

A estrutura do Sistema WebD²W é definida através de uma arquitetura, identificando cada componente, os inter-relacionamentos entre eles e suas respectivas funcionalidades.

Esta arquitetura é proposta como uma extensão de uma arquitetura genérica de data warehousing, possuindo assim novos componentes, dos quais pode-se destacar: o componente de distribuição, o componente de consulta do ambiente distribuído e o componente de manutenção do ambiente distribuído.

O sistema WebD²W define o data warehouse global como sendo o data warehouse que contém os dados que serão distribuídos para os diversos sites. De acordo com Ciferri [Ciferri02] esses dados devem ser processados antes de serem armazenados no data warehouse global. O componente de integração e extração é quem realiza a extração, tradução, limpeza e integração dos dados acessando os provedores de informação, para depois carregar o data warehouse global.

O data warehouse distribuído é formado pelo data warehouse global, que fica armazenado separadamente em um site específico, e um conjunto de unidades de distribuição, que pode ou não ser um conjunto de data marts. Essas unidades contêm os fragmentos ou réplicas do data warehouse global.

A Figura 6 mostra a arquitetura e identifica todos os componentes e o fluxo de informação entre eles [Ciferri02].

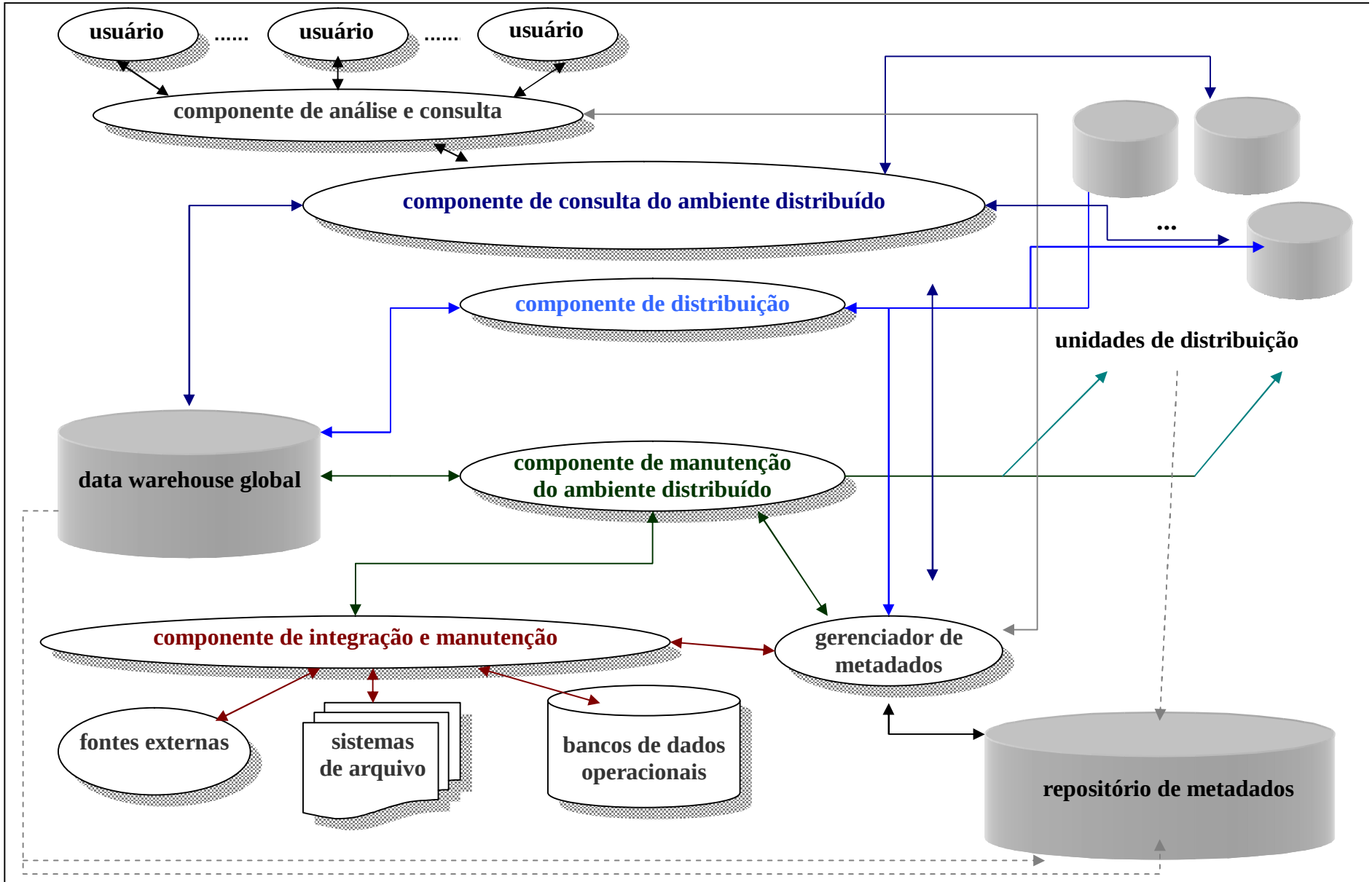

Figura 6 - Arquitetura básica do Sistema WebD²W [Ciferri02]

3.2. A importância do data warehouse global

O data warehouse global é fundamentalmente importante, pois ele contém os dados mais atualizados do sistema. A partir dele é que os fragmentos ou réplicas são gerados, e posteriormente, distribuídos nos vários sites do sistema. Ele atua como uma base intermediária entre as unidades de distribuição e os provedores de informação.

Antes dos dados serem inseridos no data warehouse global, eles são previamente extraídos, traduzidos, filtrados e integrados dos provedores de informação, como citado anteriormente. Caso a arquitetura não oferecesse suporte ao data warehouse global, seria necessário que as unidades de distribuição se comunicassem diretamente com os provedores de informação na fase de carga. Segundo Inmon [Inmon97a], diferentes unidades de distribuição poderiam ser alimentadas diretamente e individualmente pelo mesmo provedor de informação, ao passo que diferentes provedores de informação poderiam fornecer os seus dados para a mesma unidade de distribuição. Consequentemente, cada unidade de distribuição teria um conjunto de procedimentos para o seu carregamento dos dados, podendo gerar redundância e inconsistência, pois várias unidades de distribuição poderiam ter acesso aos mesmos provedores de informação. Isto resultaria em um custo adicional elevado ao sistema, devido à fase de carregamento dos dados ser um processo de atividades extremamente complexas e lentas.

A presença do data warehouse global também é importante para dar suporte à execução de consultas OLAP. Essas consultas são extremamente longas e complexas, pois manipulam uma enorme quantidade de registros durante suas execuções. Como o Sistema WebD²W possui o data warehouse global e várias unidades de distribuição, é possível que os dados sejam acessados paralelamente através da rede, acessando uma grande quantidade de dados ao

mesmo tempo, evitando-se uma sobrecarga na rede e possivelmente melhorando a performance das consultas dos usuários de SSD.

3.3. Componentes

Nesta seção, serão descritos os principais componentes da arquitetura do Sistema WebD²W e as suas respectivas responsabilidades.

3.3.1. Componente de Distribuição

Este componente realiza a fragmentação dos dados do data warehouse global e faz as alocações dos fragmentos, ou seja, faz a distribuição dos dados nos diversos sites do sistema. Tem como o principal objetivo aumentar a disponibilidade dos dados do data warehouse. Ele é composto por quatro módulos, que serão descritos abaixo.

O módulo de Requisitos tem como responsabilidade determinar um conjunto de regras que devem ser seguidas pelo projetista do data warehouse distribuído com a finalidade de definir limitações para os processos de fragmentação, replicação e alocação.

Depois dos requisitos terem sido definidos, o módulo de Fragmentação dá continuidade ao processo de distribuição dos dados, que é o objetivo principal do componente de distribuição. Assim, este módulo tem como propósito realizar o particionamento dos dados, formando porções que contêm um subconjunto de dados do data warehouse global. O particionamento pode ser tanto vertical, que consiste na fragmentação em relação aos atributos, como horizontal, que representa a fragmentação em relação às tuplas das relações de acordo com restrições estabelecidas. O algoritmo proposto neste trabalho se encontra nesse módulo, cujo objetivo é realizar a fragmentação vertical dos dados de acordo com os dados financeiros da empresa.

Já o módulo de alocação identifica quais sites do sistema devem conter quais fragmentos, ou seja, quais são os dados que devem ser alocados em cada unidade de distribuição. Estes fragmentos podem ser replicados com o objetivo de aumentar a disponibilidade dos dados, melhorando o desempenho das consultas dos usuários de SSD. Porém, a replicação dos dados requer um maior controle para manter a consistência dos mesmos.

Finalmente, o módulo de Carga é responsável pelo carregamento inicial dos dados em cada unidade de distribuição. Também realiza a gerência do processo de replicação do repositório de metadados.

A Figura 7 representa o relacionamento entre os módulos descritos acima, que pertencem ao componente de distribuição [Ciferri02].

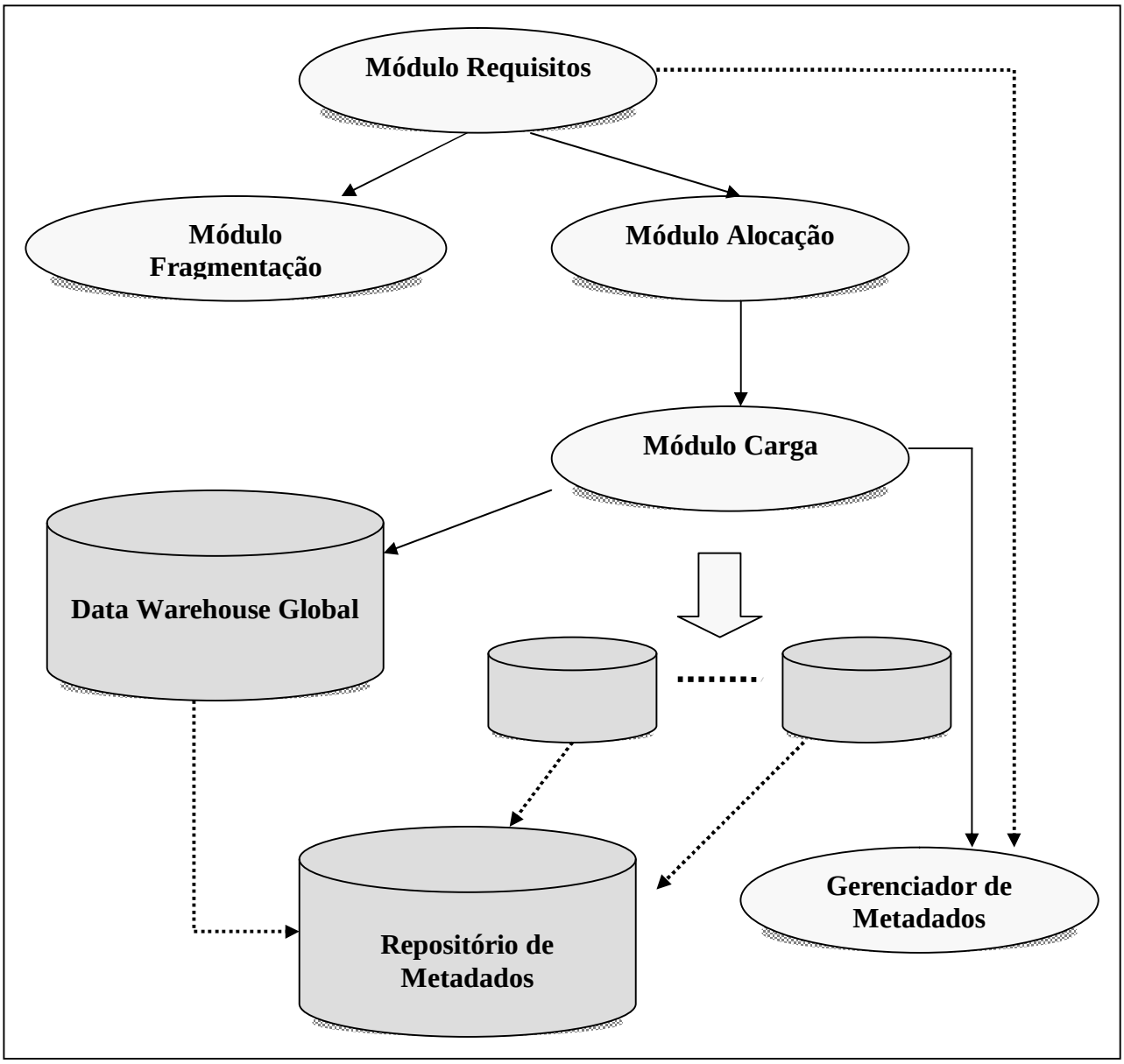

Figura 7 - Componente de distribuição [Ciferri02]

3.3.2.Componente de Consulta

Este componente pode ser representado de acordo com duas perspectivas:

- Acesso local a cada site que participa do ambiente distribuído - tem como finalidade explorar a proximidade do dado com o usuário, ou seja, o usuário pode estar localizado no mesmo site de uma determinada unidade de

distribuição ou no site do data warehouse global, podendo acessar os dados relativos às consultas no site local em que se encontra; e

- Acesso global ao ambiente distribuído como um todo: quando o usuário de SSD quer acessar o Sistema WebD²W, mas não se encontra em nenhum site da unidade de distribuição e nem no site do data warehouse global, ele deve acessar o sistema através da Internet, ou seja, o usuário se encontra em um site que não contém dados do sistema WebD²W.

Uma das principais finalidades desse componente consiste em aumentar a disponibilidade de acesso aos dados do data warehouse. Isto é possível devido aos módulos de acesso local e o de acesso global serem replicados e distribuídos em diversos sites que compõem o ambiente de data warehousing distribuído.

As oportunidades oferecidas aos usuários de SSD em realizar consultas tanto de um site de uma unidade de distribuição, quanto no site do data warehouse global ou de qualquer site que não armazene nenhum dado do Sistema WebD²W contribuem para o aumento da disponibilidade de acesso aos dados. Além disso, o acesso a esses dados deve ser de forma transparente, ou seja, o sistema deve garantir a transparência de fragmentação, replicação e localização. Os usuários devem imaginar que as consultas submetidas ao sistema são realizadas como se esse ambiente tivesse apenas um data warehouse centralizado.

Caso um usuário de SSD submeta uma consulta OLAP ao sistema WebD²W, esse componente redireciona a consulta ao site mais apropriado em respondê-la, ou simplesmente gerencia o processamento distribuído da mesma.

3.3.3.Componente de Manutenção

Tem como objetivo principal manter a consistência dos dados do data warehouse distribuído. Ou seja, este componente realiza a manutenção da consistência intra-site e entre sites.

A consistência intra-site significa manter a consistência dos dados de cada unidade de distribuição independentemente. Enquanto que a consistência entre sites visa à manutenção da consistência dos dados do data warehouse, ou seja, manter consistentes entre si os fragmentos ou réplicas através dos diversos sites do ambiente de data warehousing distribuído.

3.3.4.Componente de Integração

Este componente tem como objetivo principal realizar a integração de todos os dados que se encontram nos provedores de informações, que podem ser representados pelas fontes externas, sistemas de arquivos ou banco de dados operacionais.

Para que a integração seja realizada com sucesso, é necessário que os dados recolhidos dos provedores de informação passem pelos processos de extração, tradução, limpeza, e filtragem dos dados. Muitos problemas ocorrem quando se deseja integrar dados heterogêneos, ou seja, com vários formatos diferentes. Devido a esses problemas, este componente torna-se complexo, onde esses processos realizados são custosos e extremamente difíceis.

3.4. Ambiente Web

O data warehouse pode ser explorado e analisado via Web, trazendo várias vantagens aos usuário, devido às qualidades das características desta tecnologia. Com a utilização da Web como infra-estrutura, pode-se citar as seguintes vantagens:

- Facilidade de acesso aos dados de provedores de informação que já apresentam interface para a Web;
- Facilidade de acesso aos dados do data warehouse por usuário de SSD;
- Possibilidade de integração do ambiente informacional de empresas modernas;
- Distribuição facilitada de relatórios; e
- Redução de custos relacionados com a implantação, operação e o uso do ambiente de data warehousing.

Porém, novos desafios e dificuldades devem ser enfrentados com a utilização da Web como meio. Entre essas dificuldades, pode-se destacar as seguintes:

- Oferecer modelos de interface amigável e de fácil uso para os usuários de SSD;
- Estabelecer técnicas de segurança eficientes, como a criptografia dos dados, a utilização de firewalls, ou também programas de antivírus. Essas técnicas são necessárias para prevenir acessos indevidos, roubos e destruição de informações; e
- Aumentar a escalabilidade e a disponibilidade do sistema.

3.5. A arquitetura na Web

A arquitetura proposta pelo Sistema WebD²W, consiste em uma arquitetura de três camadas, genérica e independente da aplicação. Devido a estas características, qualquer aplicação que almeje acessar o sistema pode ser desenvolvida para ser disponibilizada via Web ou via qualquer outra arquitetura cliente-servidor.

As funcionalidades de uma aplicação podem ser agrupadas em três componentes lógicos:

- Apresentação da aplicação - define a lógica da interface do usuário. Determina funcionalidades relacionadas à apresentação dos dados na tela, quais serão as atividades de entrada e saída, valida todos os valores de entrada e faz a verificação de tipos de dados;
- Regras de negócio - define todo o processamento lógico da aplicação, ou seja, constrói as regras de negócio propriamente ditas e determina quais são as práticas administrativas da organização; e
- Gerenciamento de dados: realiza o controle da manipulação e do acesso aos dados. As funções de manipulação dos dados compõem a parte da aplicação que é responsável pelo armazenamento e pela recuperação dos dados. Enquanto que as atividades de acesso aos dados provêem o acesso físico aos dados.

Desta forma, torna-se possível a substituição de qualquer um dos componentes lógicos da aplicação sem que seja necessário modificar os demais componentes, garantindo a independência da aplicação no acesso ao data warehouse distribuído.

A Figura 8 apresenta a arquitetura em três camadas para a Web, onde a primeira camada é composta pelo cliente Web que realiza a solicitação de documentos para o servidor Web, como também formata e exibe os documentos requisitados pelos usuários. Já na segunda camada, existem dois componentes: o servidor Web e a aplicação Web, os quais respondem aos pedidos dos clientes Web, retornando os documentos requisitados. Finalmente, a última camada é composta pelo servidor de banco de dados e pelo banco de dados propriamente dito.

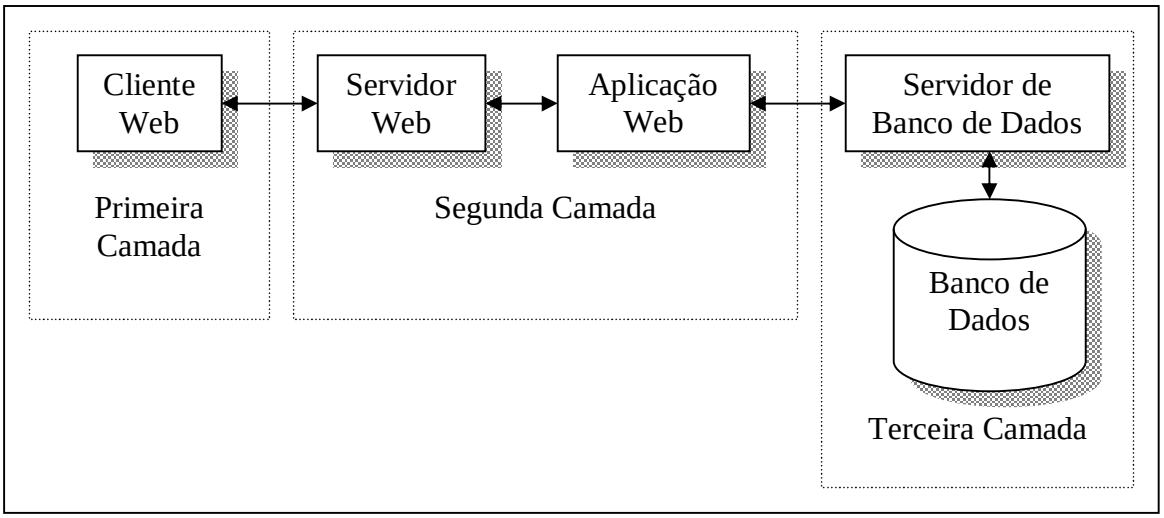

Figura 8 - Arquitetura em três camadas de integração Web com o banco de dados

3.6. Sites do Sistema WebD²W

Os usuários de SSD acessam os dados do sistema WebD²W por meio dos clientes Web através dos seguintes componentes: o site do data warehouse global, sites das unidades de distribuição e o site do sistema WebD²W para a internet. Esses sites apresentam todos os componentes destacados na arquitetura em três camadas.

A Figura 9 mostra as possibilidades de acesso ao sistema WebD²W, onde na situação (a), os clientes Web estão localizados na Intranet local e acessam o sistema a partir do data warehouse global ou de uma unidade de distribuição.

Enquanto na situação (b), os clientes acessam o sistema através da Internet, utilizando o site do sistema WebD²W para a Internet. Este site tem com objetivo principal evitar a sobrecarga de acessos externos a sites que armazenam dados do data warehouse distribuído. Ele pode ser replicado para evitar problemas de acesso a um único site do sistema WebD²W para Internet, de forma que aumentaria a disponibilidade de acesso aos dados do data warehouse distribuído.

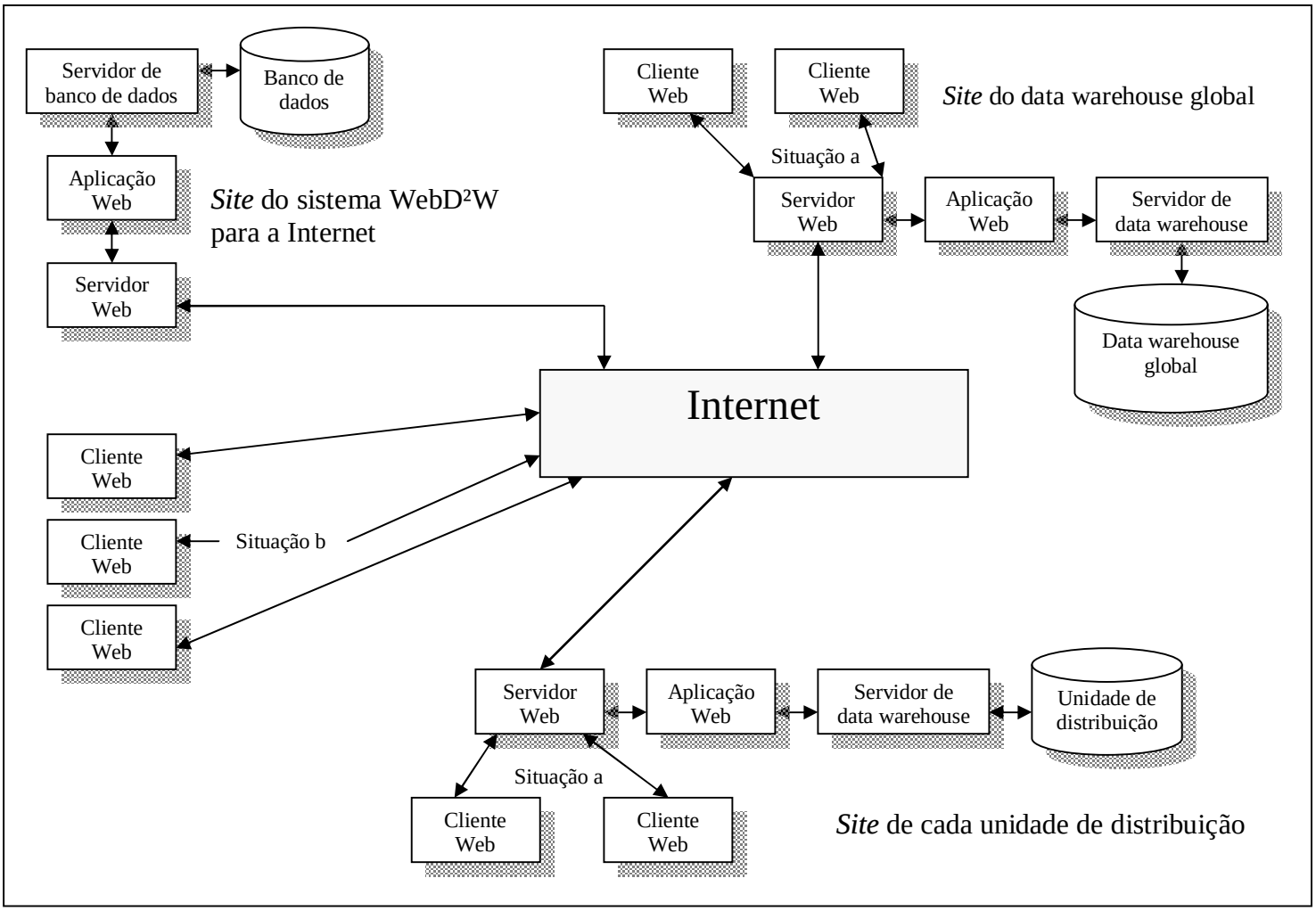

Figura 9 - Sites do sistema WebD²W [Ciferri02]

3.7. Conclusão

Este capítulo teve como objetivo apresentar o sistema WebD^{2}W, em termos da arquitetura básica do sistema. Foram introduzidos os principais componentes da arquitetura e suas respectivas funcionalidades. O componente de distribuição foi descrito com mais detalhes, pois o mesmo representa o principal componente deste sistema. Além disso, é neste componente que se encontra o módulo de fragmentação, que é responsável pela fragmentação dos dados do data warehouse global.

O capítulo 4 apresenta o algoritmo proposto neste trabalho, o qual está localizado no módulo de fragmentação. Este algoritmo realiza a fragmentação vertical dos dados em termos dos dados financeiros ou confidenciais da empresa.

4. Algoritmo de fragmentação vertical

O objetivo desse capítulo é apresentar o algoritmo proposto para realizar a fragmentação vertical dos dados baseado em grafos de derivação. Estes dados podem ser fragmentados em termos dos dados financeiros ou confidenciais da empresa, que são de extrema importância para a organização.

Este algoritmo foi fundamentado no Sistema WebD²W, onde fica localizado no componente de distribuição, ou seja, no módulo de fragmentação com o objetivo de fragmentar os dados para posteriormente serem distribuídos pelo ambiente de data warehouse distribuído. A seguir, serão explicados alguns conceitos importantes que são fundamentais para a apresentação do algoritmo proposto.

4.1. Características da fragmentação vertical

Fragmentação vertical de uma relação consiste em gerar partições da relação original contendo subconjuntos de seus atributos. O principal objetivo é aumentar o desempenho das aplicações que acessam apenas os dados contidos nos fragmentos obtidos. A fragmentação vertical foi originalmente investigada no contexto de bancos de dados centralizados, tendo como principais objetivos aumentar o desempenho das aplicações, através de uma redução no número de acessos às páginas, e identificar sub-relações mais requisitadas para serem armazenadas em subsistemas de memória mais rápidos [Özsu99].

É importante salientar a questão de replicação dos atributos que compõem a chave primária da relação. Sem isso, torna-se impossível a reconstrução da relação global original. A replicação da chave primária também ocasiona um grande benefício para o controle de integridade do banco, no entanto, pode dificultar os mecanismos de controle de concorrência.

A fragmentação vertical apresenta uma maior dificuldade de ser tratada em relação à fragmentação horizontal, devido a um número maior de alternativas disponíveis. Por exemplo, no caso da fragmentação horizontal, se o número total de predicados simples for n, existem 2n predicados *minterm* que podem ser definidos. No caso de fragmentação vertical, se uma relação possui m atributos que não são chaves primárias, o número de possíveis fragmentos é igual a B(m), onde B é o m-ésimo número de Bell. Para valores muito grandes B(m) se aproxima de mm. Com isso, torna-se inviável uma procura por soluções ótimas de fragmentação vertical, sendo necessária a investigação de heurísticas.

Existem duas abordagens heurísticas propostas para fragmentação vertical:

- Agrupamento - começa atribuindo cada atributo a um fragmento e, em cada etapa, faz-se a junção de alguns fragmentos, até que algum critério seja satisfeito; e
- Divisão - começa com uma relação e define os particionamentos benéficos com base no comportamento de acesso de aplicativos aos atributos.

4.2. Conceitos de grafos de derivação

Um grafo G (V, E) é definido pelo par de conjuntos V e E, onde:

- V - conjunto não vazio: os vértices ou nodos do grafo; e
- E - conjunto de pares ordenados e=(v,w), v e w Î V: as arestas do grafo.

Um grafo orientado, também chamado de dígrafo, representado por G (V, E) é um conjunto finito não-vazio de V e E é um conjunto de pares ordenados de elementos de vértices. E é dito conjunto de arcos ou de arestas. Um grafo

orientado é acíclico, ou seja, ele possui ciclos, como também não possui arestas múltiplas [Cormen91].

De acordo com Ciferri [Ciferri02], um grafo de derivação G consiste em um grafo orientado no qual V(G) representa um conjunto de visões agregadas (medidas numéricas agregadas), ao passo que E(V) representa um conjunto de relações de dependência < entre essas visões agregadas. Assim, se a aresta *e* pertence a G representa uma relação de dependência entre dois vértices v e u Є G de forma que v ≤ u, então inic(e) = u e term(e) = v.

A Figura 10 representa um exemplo de um grafo de derivação em termo das dimensões ncs.

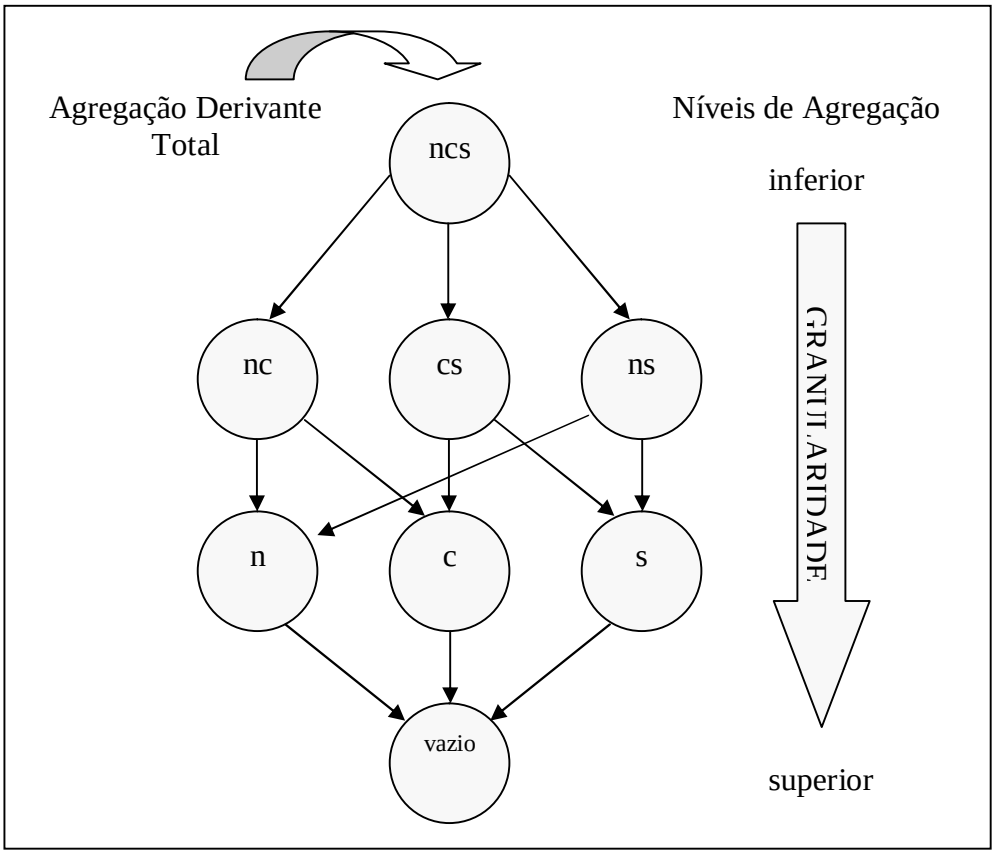

Figura 10 - Grafos de derivação em termo das dimensões ncs

4.3. Dados financeiros e confidenciais

O algoritmo proposto apresenta dois critérios para realizar a fragmentação vertical dos dados: os dados financeiros ou os dados confidenciais de uma empresa ou organização.

Os dados financeiros são de extrema importância para as organizações, pois através deles pode-se tomar decisões estratégicas em uma empresa. Por exemplo, através do dado financeiro número de vendas de um determinado produto num período específico é possível tirar conclusões sobre a viabilidade do produto.

Alguns dados financeiros são representados em termos de medidas numéricas, como por exemplo, salários dos funcionários de uma empresa, pois através da soma deles é possível calcular o custo com recursos humanos em uma organização, por exemplo. À medida que os dados vão sendo agregados, eles vão se tornando mais resumidos. Dessa forma, o lucro obtido em um produto num mês é menos agregado do que o lucro obtido em um produto durante um ano, ou seja, é mais detalhado, podendo retornar repostas mais específicas.

O Quadro 4 mostra os salários dos funcionários com os seus respectivos cargos.

Quadro 4 - Dados dos funcionários de uma empresa

Nome	Cargo	Salário (R$)
Marcos	Engenheiro	4.500
Maria	Técnico	1.200
João	Assistente	1.000
Carlos	Técnico	1.800
José	Engenheiro	5.500
Ana	Executivo	11.000
Mario	Executivo	9.000

A partir desses dados, pode ser aplicada a média dos salários dos funcionários da empresa. Dessa forma, os dados podem ser agrupados, ou seja, resumidos fornecendo dados com menos detalhes. Agrupando os dados em cargo, tem-se a média dos salários por cargo. O Quadro 5 mostra o resultado obtido.

Quadro 5 - Médias dos salários dos cargos de uma empresa

Cargo	Salário (R$)
Executivo	10.000
Engenheiro	5.000
Técnico	1.500
Assistente	1.000

Os dados confidenciais também são importantes em uma organização. Por exemplo, no contexto de um hospital, os resultados dos exames dos pacientes devem ser confidenciais.

Assim, um paciente poderia realizar um exame de AIDS, e o resultado obtido deve ser restrito ao mesmo. Com a fragmentação vertical dos dados dos pacientes, poderia ser realizada uma análise em termos da ocorrência dos

resultados obtidos nos exames, ou seja, o número de pacientes infectados com o vírus HIV. O Quadro 6 mostra os atributos de um paciente, resumidamente, de forma a esclarecer a necessidade da fragmentação vertical dos dados.

Quadro 6 - Dados de pacientes resumidos de um hospital

Nome	Exame	Resultado
Marcos	HIV	Não infectado
Maria	Tipo Sanguíneo	A^+
João	Colesterol	Taxa Elevada
Carlos	HIV	Infectado
José	Tipo Sanguíneo	O^-

Com a fragmentação vertical dos dados mostrados no Quadro 6 acima em termos dos dados confidenciais do paciente, ou seja, do tipo de exame com o seu respectivo resultado, tem-se os dados mostrados no Quadro 7.

Quadro 7 - Dados de pacientes fragmentados em termos dos dados confidenciais

Exame	Resultado
HIV	Não infectado
Tipo Sanguíneo	A^+
Colesterol	Taxa Elevada
HIV	Infectado
Tipo Sanguíneo	O^-

Através da fragmentação é possível determinar o tipo de exame que é mais realizado no hospital, como também saber a porcentagem de pacientes infectados com o vírus HIV. Outro tipo de análise que poderia ser realizada era saber a quantidade de pacientes que estão com o nível de colesterol elevado, permitindo fazer um estudo comparativo entre os dados obtidos dos anos anteriores.

4.4. Algoritmo de fragmentação vertical proposto

Nesta seção, será apresentado o algoritmo de fragmentação vertical utilizando grafos de derivação. Este algoritmo realiza a fragmentação dos dados de acordo com as dimensões que podem representar dados financeiros ou confidenciais da organização.

4.4.1.Entradas do algoritmo FV

A seguir serão apresentadas as entradas do algoritmo de fragmentação vertical.

Entradas:

- O grafo de derivação G a ser fragmentado, representado pelo esquema do data warehouse global; e
- O conjunto de dimensões a serem fragmentadas, que podem ser representadas, por exemplo, em termos de dados financeiros ou confidenciais.

O grafo de derivação total, que representa o data warehouse global, é representado por G (V, E), onde:

- V (G) = $\{v^1, v^2, ..., v^n\}$, n Є N*
- E (G) = $\{e^1, e^2, ..., e^m\}$, onde m Є N.

O primeiro vértice representa a agregação derivante total, ou seja, a partir de v^1 é possível derivar todas as dimensões do data warehouse. Logo, a partir de v^1, o algoritmo proposto vai realizar a fragmentação e como resultado, ele produzirá k novos grafos de derivação ($G^1, G^2, ..., G^k$) de acordo com o número de dimensões a serem fragmentadas. À medida que o vértice v^1 vai sendo derivado,

ele vai gerando descendentes mais agregados até chegar ao vértice v^n, que corresponde ao vazio, ou seja, o mais agregado, que representa o número de ocorrências.

Como exemplo, para o grafo representado na Figura 10:

- V(G) = {ncs, nc, cs, ns, n, c, s, vazio}
- E(G) = {ncs_nc, ncs_cs, ncs_ns, nc_n, nc_c, cs_c, cs_s, ns_n, ns_s, n_vazio, c_vazio, s_vazio}

O conjunto de dimensões a serem fragmentadas é representado por $CD^F = \{df^1, df^2, df^3, ..., df^k\}$, onde $k \in N^*$.

A função de agregação das dimensões fornecida pelo algoritmo consiste na função que determina as suas respectivas ocorrências.

4.4.2.Detalhamento do algoritmo FV

Será apresentado o processo de fragmentação do algoritmo, como também os demais processos pertencentes ao mesmo. No final, após o processo de fragmentação, o algoritmo constrói as arestas, ou seja, as relações de dependências existentes entre as agregações.

O Quadro 8 apresenta o pseudocódigo do algoritmo proposto.

Quadro 8 - Pseudocódigo do algoritmo de fragmentação vertical

Algoritmo FV

Entradas:

➡ **G(V,E);**

// o grafo de derivação a ser fragmentado, onde:
// os vértices são: V(G) = {v^1, v^2,..., v^n}, onde n ∈ N*,
// onde v^1 representa a agregação derivante total.
// as arestas são: E(G) = {e^1, e^2, ..., e^m}, onde m ∈ N.

➡ **CD^F = {df^1, df^2, ..., df^w}, onde w ∈ N*;**

// o conjunto de dimensões a serem fragmentadas.
// Estas dimensões podem, por exemplo, representar
// dados financeiros ou confidenciais da empresa.
// Devido aos dados financeiros ou confidenciais não
// poderem ser todas às vezes representados em termos
// de medidas numéricas, o algoritmo proposto não
// recebe como entrada um conjunto de funções de
// agregação, e sim, define como padrão a função de
// agregação de ocorrências. No caso do exemplo mostrado
// nos Quadros 6 e 7, o algoritmo determina a ocorrência
// de pacientes que estão infectados com o vírus HIV,
// ou seja, determina o número de pacientes que estão
// infectados, pois não tem sentido aplicar uma função
// de agregação, por exemplo, de média nesse contexto.

Saídas:

➪ **CG = {G1(V^1, E^1),...,Gj (V^k, E^k),...,G^k (V^w, E^w)}, onde $1 \leq k \leq w$;**

// conjunto de k grafos de derivação fragmentados
// de acordo com as dimensões passadas como parâmetro.
// Esses fragmentos são representados em diferentes
// níveis de agregação de suas dimensões.
// Cada grafo consiste numa visão materializada.

// Parte da execução do algoritmo.

Início:

// primeiramente, faz-se o vértice inicial v seja o
// v1, ou seja, o vértice de agregação derivante total.
v ← v^1;
// para cada vértice a ser analisado, verifica se
// o mesmo contém apenas as dimensões que pertencem
// a CD^F. Caso ele contenha, este vértice
// será fragmentado, gerando os seus níveis de
// agregação. Como também, o algoritmo
// verifica se o vértice analisado contém
// todos as dimensões que contém em v^1 menos
// as dimensões que pertencem a CD^F. No exemplo
// dos Quadros 6 e 7, seriam gerados os
// fragmentos que contêm os nomes dos pacientes
// e outro com os exames e resultados.
// Dessa forma, o algoritmo gera dois grafos
// representando os fragmentos.
// A partir desses grafos, o algoritmo gera os
// grafos de reconstrução que contêm
// os diferentes níveis de agregação da dimensão.

```
// Nessa fase, a seguir, são gerados dois fragmentos,
// um que contém todas as dimensões que pertencem
// a CD^F, e outro que contém todas as dimensões
// do vértice inicial e não contém as dimensões de CD^F.
Se v contém apenas CD^F
     Então // fragmente o vértice
          f^k ← v fragmentado com apenas as dimensões
                 de CD^F;
          f^k + 1 ← v fragmentado com todas as dimensões
                  de v^1 menos as de CD^F;
Fim Se

// para cada fragmento gerado, instanciá-lo
// em um vértice e fazer a associação do mesmo à
// um grafo de derivação G^k.
Repita
     Faça
     // é criado um vértice, instanciado e associado
     // com o fragmento que está sendo
     // analisado no momento.
     instanciar v^k com f^k;
     associar v^k à V^k(G^k);
Até não existirem fragmentos

// Depois de criar os fragmentos, é necessário
// reconstruir, ou seja, realizar o processo de
// reconstrução. Este processo é responsável por
// garantir que os fragmentos contenham todas
// as agregações, obtendo todas as formas de
// combinações de agregação em função das dimensões
// do conjunto CD^F.
```

```
Repita
    Faça
    // gera-se um conjunto de vértices derivados
    // do grafo fragmentado a partir do vértice v^k.
      CV ← adicione os vértices que representam as
      combinações das dimensões de v^k onde dimensão
      m^k está presente;
    // enquanto existirem vértices no conjunto
    // agregado.
    Repita
        Faça
        // associa o vértice v^k como ancestral ao
        // vértice agregado em questão.
        associe v^k como ancestral direto (pai) do
        vértice v^z do conjunto de vértices CV;
        // insere os valores no vértice em questão
        // de acordo com seu ancestral v^k e
        // a função de agregação
        // que representa o número de ocorrências.
        determinar os valores
        a serem armazenados em v^z
        de acordo com seu ancestral v^k e com f_ag(m^k);
        // associa o vértice agregado ao
        // grafo fragmentado G^k.
        associar v^z à V^z(G^k);
    Até não existirem vértices em CV

  // Refaz todo o procedimento de criação
  // do conjunto de vértices para cada descendente
  // que representa a combinação das dimensões,
  // até que não existam mais descendentes.
  Repita
```

```
                refaça o procedimento de criação do conjunto
                de vértices derivados para cada descendente
                do conjunto CV até que não existam mais
                descendentes;
        Até não existirem vértices fragmentados v^k

        // Depois de realizar a fragmentação e fazer todas
        // agregações, é necessário criar as arestas, ou seja,
        // as dependências entre as agregações criadas.
        // O conjunto de agregações geradas é representado por:
        // G^1,... G^k,... G^w.
        Repita
          // cria a aresta e associa ao grafo em questão.
          crie uma aresta, de forma idêntica a e^k e
          associe a E^k(G^k);
        Até não existirem agregações G^k


// Fim do Algoritmo de Fragmentação baseado
// em grafos de derivação.
Fim
```

4.4.3.Exemplo de aplicação do algoritmo

Nesta seção, será mostrado um exemplo utilizando o software desenvolvido com o propósito de implementar o algoritmo proposto neste trabalho. Este software foi implementado através do JBuilderX utilizando a linguagem de programação Java, devido as suas vantagens, como: portabilidade, segurança e eficiência.

Este software apresenta uma interface amigável e simples, onde o usuário pode visualizar o grafo de derivação gerado a partir das dimensões fornecidas, como também, o script de geração de tabelas correspondente ao conjunto de vértices que fazem parte do conjunto de dimensões a serem fragmentadas.

Devido à necessidade de proteger os dados de uma organização, o exemplo a ser utilizado nessa seção representa uma fragmentação vertical de uma tabela simplificada correspondente a um paciente de um hospital. O paciente possui os seguintes atributos: nome, convênio, exame, resultado e prontuário. A fragmentação será realizada em termos de dados confidenciais, ou seja, dos dados: exame, resultado e prontuário, que são informações restritas ao paciente, e logicamente, para a equipe médica que tem acesso a essas informações. Esta fragmentação permite aumentar a segurança desses fragmentos que contêm os dados confidenciais, que são extremamente importantes.

Como resultado, será gerado um grafo de derivação a partir da agregação derivante total, que no caso será: “ncerp”, onde “n” representa o nome, “c” o convênio, “e” o exame, “r” o resultado e “p” o prontuário. Serão mostrados os scripts para a criação de tabelas em SQL dos vértices que estiverem na cor verde, isto porque eles representam os fragmentos que contêm as dimensões correspondentes aos dados confidenciais, como também, os que contêm todas as dimensões da agregação derivante total menos o conjunto de dimensões a serem fragmentadas.

A Figura 11 representa a fragmentação feita para o exemplo proposto, onde é gerado um fragmento com as dimensões “nc” e outro com as dimensões “erp”.

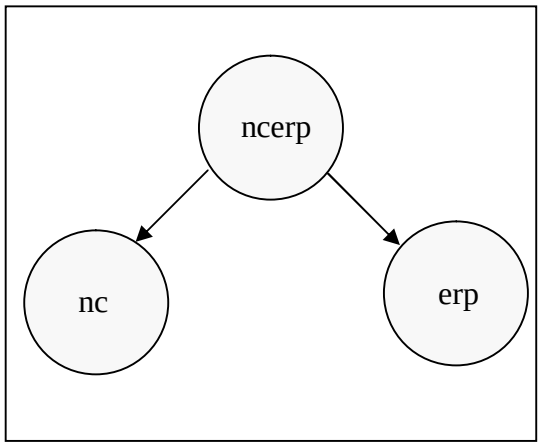

Figura 11 - Fragmentação resultante do exemplo aplicado

A partir disso, são gerados todas as agregações de cada fragmento até chegar ao vértice vazio, que representa o maior nível de agregação. A agregação é realizada de acordo com a medida numérica de ocorrência.

A Figura 12 e Figura 13 representam os grafos de derivação de cada fragmento gerado.

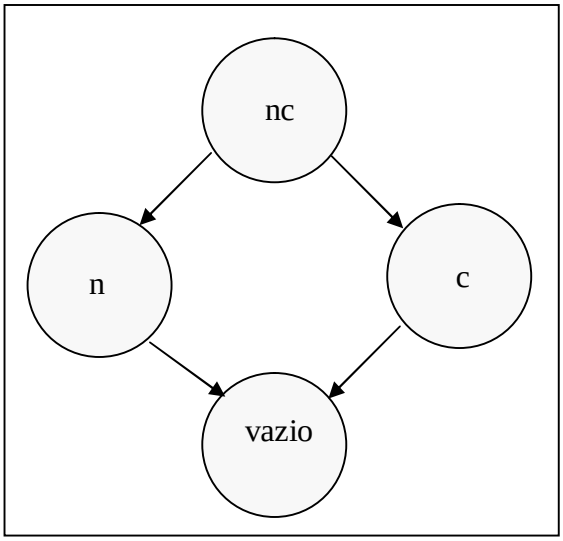

Figura 12 - Grafo de derivação do fragmento “nc”

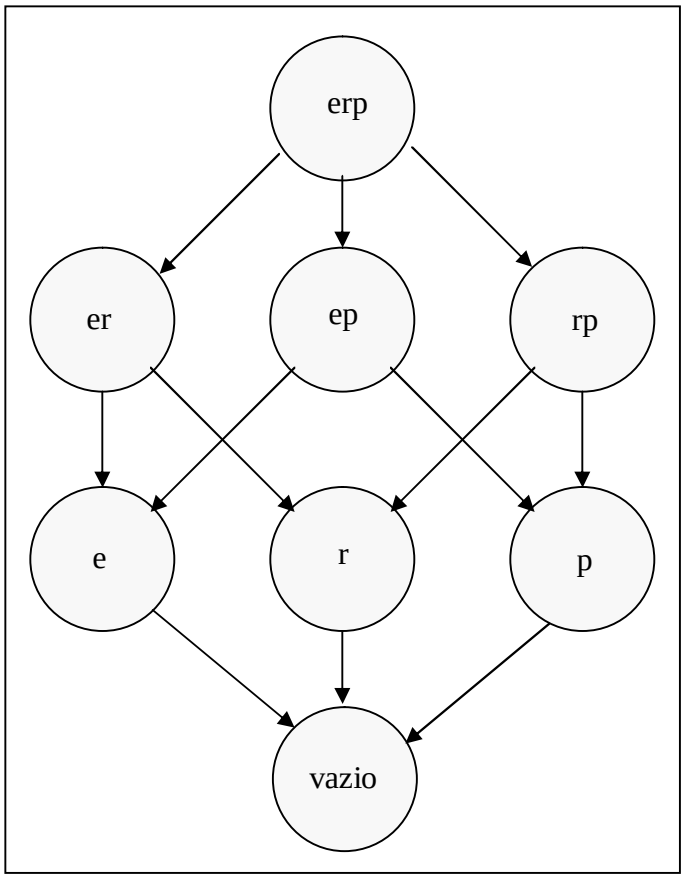

Figura 13 - Grafo de derivação do fragmento “erp”

A Figura 14 mostra o *screenshot* do grafo de derivação obtido nessa fragmentação.

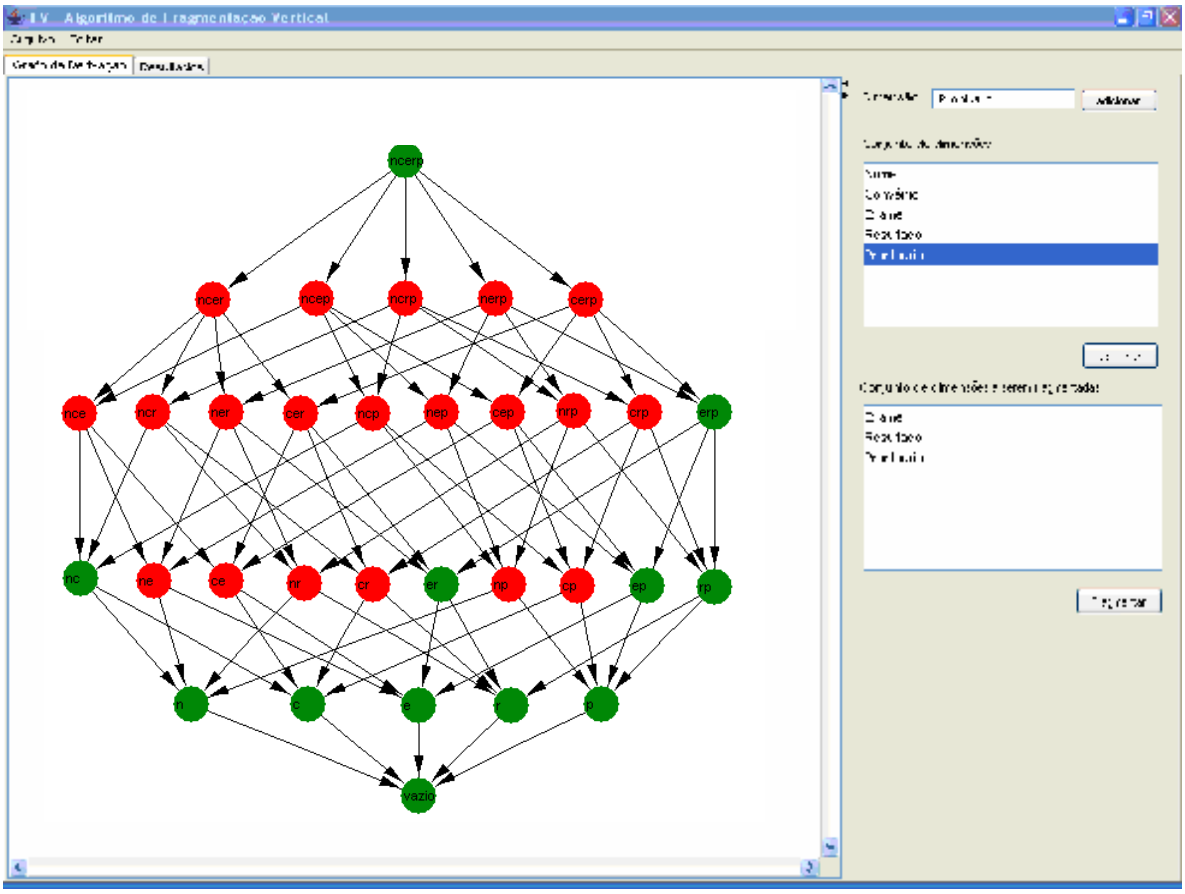

Figura 14 - *Screenshot* do software que realiza a FV do exemplo

A partir da Figura 14, pode-se perceber que são gerados os descendentes da agregação derivante total até se chegar aos vértices que contém “erp” e no vértice que contém “nc”. Estes vértices representam os vértices fragmentados, onde ele e os seus descendentes são marcados com a cor verde.

Depois de determinar os dois vértices fragmentados, é preciso realizar a agregação dos valores. Essa agregação se faz através de uma função de agregação, que no caso do algoritmo proposto, é realizada por verificação de ocorrência. Assim, o último vértice vazio corresponde ao maior nível de agregação. Este vértice pode representar, por exemplo, o número de pacientes infectados com o vírus HIV, o tipo de exame que está sendo mais realizado e o tipo sanguíneo que é mais comum.

A Figura 15 apresenta o script gerado para os vértices fragmentados.

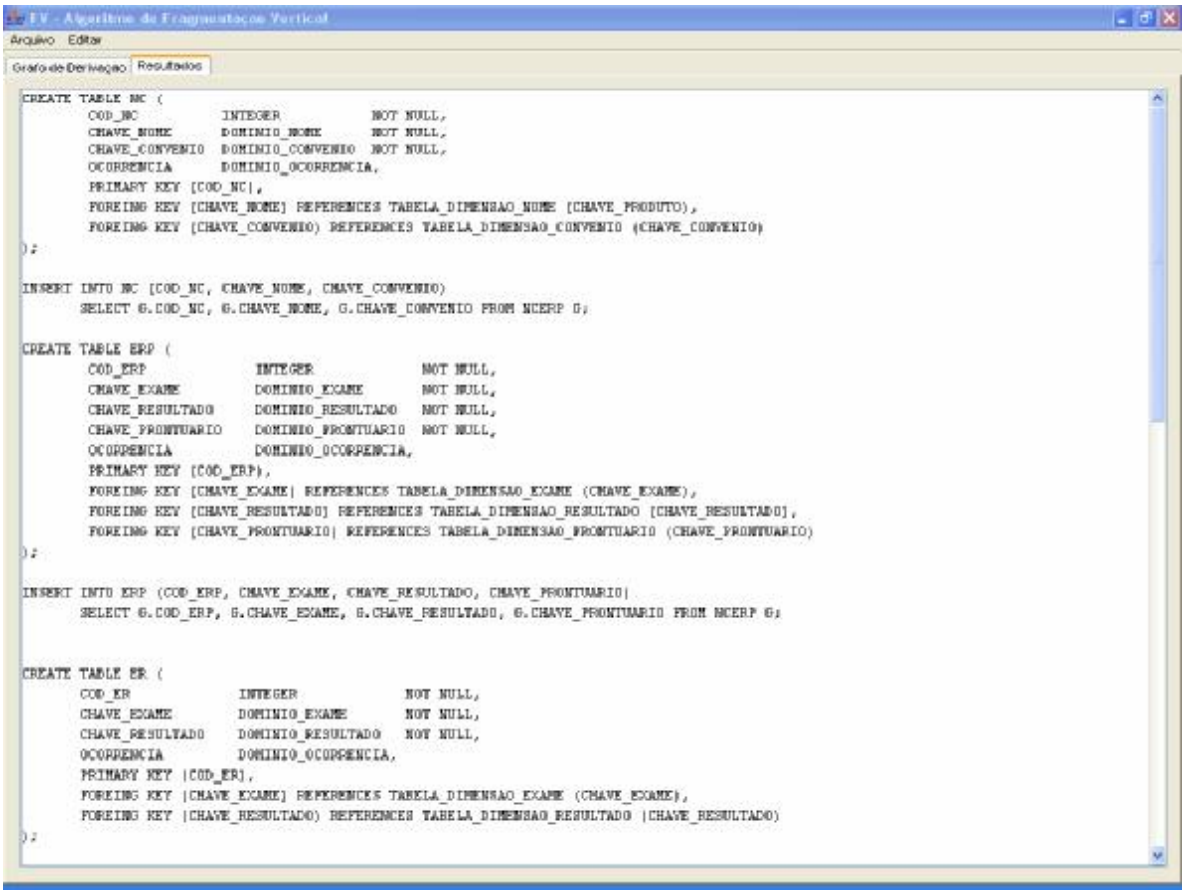

Figura 15 - GUI do script gerado pelo software do exemplo proposto

4.4.4. Vantagens e desvantagens do algoritmo

A principal vantagem oferecida pelo algoritmo proposto consiste na segurança, de forma que os fragmentos gerados em relação aos dados financeiros ou confidencias de uma empresa podem ser alocados em *sites* extremamente seguros. Alem disso, por haver um ganho de performance em relação ao acesso aos dados, pois os dados podem ser alocados em *sites* diferentes, possibilitando o acesso paralelo aos mesmos.

Porém, este algoritmo apresenta a desvantagem de não analisar os dados que estão sendo utilizados mais frequentemente. Consequentemente, um conjunto de dados que está sendo acessado com muita freqüência pode ser fragmentado, necessitando reunir esses dados quando forem requisitados.

5. Conclusões e Trabalhos Futuros

Este trabalho teve como objetivo o desenvolvimento de um algoritmo de fragmentação vertical de dados, baseado em grafos de derivação, com o intuito de aumentar a segurança dos dados relevantes de uma organização, ou seja, de dados financeiros ou confidenciais.

De acordo com Ciferri [Ciferri02], data warehouse representa uma única base de dados centralizada. Distribuir os dados armazenados nessa base de dados levando-se em consideração as características intrínsecas de aplicações de data warehousing apresenta várias vantagens, porém introduz novos desafios a ambientes de data warehousing. Dentro deste contexto, foi proposto o sistema WebD²W que enfoca a distribuição dos dados do Data Warehousing.

Dessa forma, foi utilizada como base a arquitetura proposta por Ciferri [Ciferri02], onde o componente principal é o componente de distribuição. É nele que este algoritmo se encontra, pois é através do módulo de fragmentação, presente neste componente, que a fragmentação dos dados é realizada. Após a fragmentação, os dados são distribuídos com o objetivo de aumentar a disponibilidade e o acesso aos mesmos.

O resultado obtido neste trabalho foi além do desejado, pois além de toda fundamentação teórica do algoritmo, foi desenvolvido um programa capaz de realizar o algoritmo proposto.

Em continuidade ao trabalho desta pesquisa, recomenda-se o desenvolvimento de um algoritmo que aborde os dois tipos de fragmentação (misto), ou seja, um algoritmo que realize a fragmentação horizontal dos dados, de acordo com um conjunto de restrições, e a fragmentação vertical, analisando os dados que estão sendo acessados mais frequentemente. O objetivo principal seria

aumentar a disponibilidade dos dados, como também, a performance no acesso aos dados.

Referências Bibliográficas

1. [Barquini96] BARQUINI, Ramon, Planning and Designing the Warehouse, New Jersey, Prentice-Hall, 1996.

2. [Ciferri02] CIFERRI, C.D.A., Distribuição dos Dados em Ambientes de Data Warehousing: O Sistema WebD²W e Algoritmos Voltados à Fragmentação Horizontal dos Dados, Universidade Federal de Pernambuco, 2002.

3. [Cormen91] Th.H. Cormen, Ch.E. Leiserson, R.L. Rivest, C. Stein Introduction to Algorithms, 2nd edition, MIT Press & McGraw-Hill, 1991.

4. [Diestel97] DIESTEL, R. Graph Theory. Springer-Verlag New York, Inc., USA, 1997. 286 pp. ISBN 0-387-98211-6.

5. [Griffin] GRIFFIN, J., The Pitfalls of “Virually” Building a Data Warehouse, acessada em 2004, disponível em <http://www.datawarehouse.com>.

6. [Hackney98] HACKNEY, Douglas. Data Warehouse Delivery, 1998.

7. [Hammer79] HAMMER, M., Niamir, B., A Heuristic Approach to Attribute Partitioning, ACM, 1979.

8. [Inmon97a] INMON, William H.,Como Construir o data warehouse. Rio de Janeiro: Campus, 1997a.

9. [Inmon97b] INMON, William H., HACKARTHORN, Richard D. Como usar o data warehouse. Rio de Janeiro: IBPI Press, 1997b.

10. [Inmon99] INMON, W. H. et al. Gerenciando Data Warehouse, São Paulo: Makron Books,1999.
11. [Inmon01] INMON, W. H. Building the Corporate Information Factory from a Blueprint, 2001. Disponível em: http://www.billinmon.com. Acesso em 2005.

12. [Kimball96] KIMBALL, Ralph, The Data Warehouse Toolkit: practical techniques for building Dimensional Data Warehouse – John Wiley & Sons, Inc. 1996.

13. [Kimball98a] KIMBALL, R. The Data Warehouse Lifecycle: Expert Methods for Designing, Developing and Deploying Data Warehouses. New York John Wiley & Sons Inc., 1998a.

14. [Kimball98b] KIMBALL, R. Data Warehouse Toolkit: Técnicas para Construção de Data Warehouse Dimensionais. São Paulo: Makron Books, 1998b.

15. [Kimball00] KIMBALL, R., MERZ, R., Data Webhouse: Construindo o Data Warehouse para Web. Rio de Janeiro: Campus, 2000.

16. [Melo97] MELO, Rubens Nascimento. Data Warehousing (Tutorial). In: XIII SBBD - Simpósio Brasileiro de Banco de Dados, Salvador, 1997.

17. [Navathe84] NAVATHE, S., Ceri S., Wiederhold G., Dou, J., Vertical Partitioning of Algorithms for Database Design, ACM Trans, 1984.

18. [Özsu99] ÖZSU, M.T., Valduriez, Principles of Distributed Database Systems, pages 108 - 165, Upper Saddle River, New Jersey, USA. Prentice-Hall, 1999.

19. [Singh97] SINGH, H., Data Warehousing: Concepts, Technologies, Implementations, and Management, Upper Saddle River, New Jersey, USA. Prentice-Hall, 1997.

20. [Singh01] SINGH, H. S. Data Warehouse. Conceitos, Tecnologias, Implementação e Gerenciamento. Markron Books, 2001.

21. [White] White, C., Managing Data Warehouse Meta Data, DM Review, acessada em 2004, disponível em <http://www.dmreview.com>.