

**NILO JOÃO CUOCHI MELHORANÇA JÚNIOR**

**BANCO DE DADOS DISTRIBUÍDOS PARA AUXILIAR NA GERÊNCIA  
DE REDES**

Florianópolis  
2002

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**NILO JOÃO CUOCHI MELHORANÇA JÚNIOR**

**BANCO DE DADOS DISTRIBUÍDOS PARA AUXILIAR NA GERÊNCIA  
DE REDES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre, pelo Programa de Pós-Graduação de Ciências da Computação – Curso de Ciências da Computação, da Universidade Federal de Santa Catarina.

Orientador: Prof. Dr. Carlos Becker Westphall

Florianópolis, Dezembro de 2002.

# **BANCO DE DADOS DISTRIBUÍDOS PARA AUXILIAR NA GERÊNCIA DE REDES.**

**Nilo João Cuochi Melhorança Júnior**

Esta dissertação foi julgada como adequada para a obtenção do título de mestre em Ciência da Computação – Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

---

Prof. Dr. Fernando A. O. Gauthier  
Coordenador do curso

Banca Examinadora:

---

Prof. Dr. Carlos Becker Westphall  
Orientador

---

Prof. Dr. Vitório Bruno Mazzola

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Carla Merkle Westphall

---

Prof . Dr. José Mazzucco Jr.

Aos meus avós, Nilo Tranquilo Melhorança e Jandira Coughi Melhorança.

## **AGRADECIMENTOS**

Agradecimentos especiais ao Prof. Carlos Becker Westphall, pela orientação e apoio valiosos.

Aos meus pais Nilo e Elizabeth pelo apoio e incentivo dado durante todo o decorrer do curso do mestrado.

A minha namorada Aline pela paciência, carinho e pela compreensão.

À funcionária da secretaria de CPGCC Valdete pela atenção dispensada.

Aos demais membros da banca examinadora pelas sugestões feitas na apresentação do trabalho individual e pelo apoio dado ao desenvolvimento deste trabalho.

## RESUMO

Atualmente a maioria das empresas de médio e grande porte formadas de filiais que geralmente se localizam geograficamente distantes entre si não possuem uma integração entre seus bancos de dados, ou seja, armazenam e consultam apenas os dados locais. Isso é indesejável não somente para informações financeiras e contábeis, mas também no âmbito de gerência de redes. Devido ao crescimento do número de tecnologias de acesso a Internet que usam banda larga e por conseqüência seu barateamento, surge a possibilidade de utilização destes serviços nas empresas. Assim como o crescimento do número de softwares de banco de dados com suporte a distribuição de informações (e seu barateamento) facilita a integração de dados. O emprego de banco de dados distribuídos e gerência de redes de forma conjunta torna as informações de gerência (além de dados diversos das filiais) mais integradas e fáceis de se consultar e analisar.

## **ABSTRACT**

Nowadays the most of the medium and big companies formed of branches that are generally located geographically distant don't possess an integration between your databases, that is to say, they store and just consult only the local database. That is undesirable not only for financial and accounting information, but also in the ambit of network management. With the growth of the number of internet access technologies that use wide band and for its consequence your depreciate cost, it turns the use of these services more accessible to the companies. As well as the growth of the number of database softwares with support the distribution of information facilitate the integration of yours databases. Then, if use distributed database and network management together would turn the network management data (and all others information about the company) more integrated and too easy to consult and analyze.

## SUMÁRIO

<b>LISTA DE FIGURAS</b>	<b>11</b>
<b>LISTA DE SIGLAS</b>	<b>12</b>
<b>1 INTRODUÇÃO</b>	<b>14</b>
1.1 Objetivos	15
1.2 Trabalhos Correlatos	16
1.3 Organização do Trabalho	18
<b>2 BANCO DE DADOS DISTRIBUÍDOS</b>	<b>19</b>
2.1 INTRODUÇÃO	19
2.2 O QUE É BANCO DE DADOS DISTRIBUÍDO?	19
2.3 CARACTERÍSTICAS DE UM BANCO DE DADOS DISTRIBUÍDOS	20
2.4 ARQUITETURA DE UMA BANCO DE DADOS DISTRIBUÍDOS	22
2.5 PROJETO DE UMA BANCO DE DADOS DISTRIBUÍDOS	23
2.5.1 Projeto de fragmentação de dados	24
2.5.2 Projeto de replicação de dados	24
2.5.3 Projeto de alocação de dados	26
2.6 ARQUITETURA DE UM SGBDD	26
2.6.1 Gerência de transações	26
2.6.2 Controle de concorrência	27
2.6.3 Gerência de recuperação	27
2.6.4 Integridade	28
2.6.5 Segurança	28
2.6.6 Administração de Bancos de Dados Distribuídos	29
<b>3 GERÊNCIA DE REDES</b>	<b>30</b>
3.1 INTRODUÇÃO	30
3.2 O QUE É GERÊNCIA DE REDES?	30



3.3	ÁREAS DA GERÊNCIA DE REDES	31
3.4	MODELO DE GERÊNCIA BASEADO NO SNMP	32
3.5	FORMATO DAS INFORMAÇÕES GERENCIÁVEIS	33
3.6	NOMEAÇÃO DE VARIÁVEIS SNMP	34
3.7	PROTOCOLO DE GERENCIAMENTO	35
3.8	FLUXO DE OPERAÇÕES	35
3.9	COMUNICAÇÃO NO PROTOCOLO SNMP	36
3.10	FERRAMENTAS AUTOMATIZADAS PARA GERÊNCIA	37
<b>4</b>	<b>BANCO DE DADOS DISTRIBUÍDOS EM ORACLE</b>	<b>39</b>
4.1	INTRODUÇÃO	39
4.2	ORACLE COMO SGBDD	39
4.3	TIPOS DE <i>DATABASE LINKS</i> USADOS NO ORACLE	40
4.4	CARACTERÍSTICAS DO ORACLE	41
4.5	DISTRIBUIÇÃO DE DADOS NO ORACLE	43
<b>5</b>	<b>IMPLEMENTAÇÃO E DESENVOLVIMENTO DA INTEGRAÇÃO DE BDD E GERÊNCIA DE REDES</b>	<b>46</b>
5.1	INTRODUÇÃO	46
5.2	SITUAÇÃO ATUAL	47
5.3	PROPOSTA DE INTEGRAÇÃO DE BANCOS DE DADOS	48
5.4	VANTAGENS DO BANCO DE DADOS DISTRIBUÍDO	49
5.5	DESvantagens DO BANCO DE DADOS DISTRIBUÍDO	50
5.6	VANTAGENS DA INTEGRAÇÃO ENTRE BDD E GERÊNCIA DE REDES	51
5.7	AMBIENTE DE IMPLEMENTAÇÃO	52
5.8	MODELO DE DADOS DOS SÍTIOS	53
5.9	INTEGRAÇÃO DOS DADOS DENTRO DA FILIAL	55
5.10	DISTRIBUIÇÃO DOS DADOS	56
5.11	MODELO DE DADOS (DEPOIS DA DISTRIBUIÇÃO DE DADOS)	57
5.12	CONEXÃO ENTRE AS FILIAIS	59
5.13	REPLICAÇÃO DOS DADOS	61
5.14	FUNCIONAMENTO DO PROCESSO DE GERÊNCIA	64
5.15	FUNCIONAMENTO E VALIDAÇÃO DO PROTÓTIPO	65

5.16	VALIDAÇÃO DO PROCESSO DE OBTENÇÃO DOS DADOS DE GERÊNCIA	67
5.17	TESTES REALIZADOS	69
5.18	RESULTADOS OBTIDOS	72
<b>6</b>	<b>CONCLUSÃO</b>	<b>74</b>
6.1	PRINCIPAIS CONTRIBUIÇÕES	76
6.2	TRABALHOS FUTUROS	77
	<b>REFERÊNCIAS</b>	<b>78</b>

## LISTA DE FIGURAS

<b>Figura 1 -</b>	Banco de dados distribuído	20
<b>Figura 2 -</b>	Banco de dados centralizado em uma rede de computadores	20
<b>Figura 3 -</b>	Réplica periódica	25
<b>Figura 4 -</b>	Réplica contínua	25
<b>Figura 5 -</b>	Modelo de gerência baseado no SNMP	33
<b>Figura 6 -</b>	Árvore de identificadores de objetos	34
<b>Figura 7 -</b>	Interação de uma requisição SNMP	36
<b>Figura 8 -</b>	Situação atual	47
<b>Figura 9 -</b>	Ambiente de Implementação	52
<b>Figura 10 -</b>	Modelo de dados dos sítios (antes da distribuição dos dados)	53
<b>Figura 11 -</b>	Modelo de dados dos sítios (depois da distribuição dos dados)	57
<b>Figura 12 -</b>	Interface de criação de <i>database link</i>	60
<b>Figura 13 -</b>	Funcionamento do processo de gerência	64
<b>Figura 14 -</b>	Tráfego de dados na matriz	70
<b>Figura 15 -</b>	Tráfego de dados na filial de Rio do Sul	70
<b>Figura 16 -</b>	Tráfego de dados na filial de Joinville	71

## LISTA DE SIGLAS

ANSI	<i>American National Standards Institute</i>
ASN.1	<i>Application Programming Interface</i>
BD	Banco de Dados
BDD	Banco de Dados Distribuído
CMIP	<i>Common management Information Protocol</i>
CPU	<i>Central Processing Unit</i>
DBA	<i>DataBase Administrator</i>
DDG	Diretório de Dados Global
DES	<i>Data Encryption Standard</i>
DRDA	<i>Distributed Relational Database Architecture</i>
E/S	Entrada / Saída
GD	Gerente de Dados
GT	Gerente de Transações
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
MIB	<i>Management Information Base</i>
NMS	<i>Network Management Station</i>
ODBC	<i>Open DataBase Connectivity</i>
OID	<i>Object Identifier</i>
OSI	<i>Open Systems Interconnection</i>
PDU	<i>Protocol Data Unit</i>
PKC	<i>Public Key Cryptosystems</i>
PL/SQL	<i>Procedural Language / Structured Query Language</i>
SBDD	Sistema de Banco de Dados Distribuído
SGBD	Sistema Gerenciador de Banco de Dados Centralizado
SGBDD	Sistema Gerenciador de Banco de Dados Distribuído
SNMP	<i>Simple Network management Protocol</i>
SQL	<i>Structured Query Language</i>

TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
WAN	<i>Wide Area Network</i>
WWW	<i>World-Wide Web</i>
XML	<i>Extensible Markup Language</i>

## **1 INTRODUÇÃO**

A utilização efetiva de redes de computadores e banco de dados nas diversas áreas de negócios tem sido um fator determinante para o sucesso das empresas da atualidade, automatizando e organizando diversas atividades e servindo como um eficiente meio de comunicação.

A mais de quatro anos trabalho com análise e desenvolvimento de softwares para empresas de médio e grande porte. A grande maioria destes softwares utiliza banco de dados para armazenar as informações destas corporações e boa parte destas instituições usam técnicas de gerência de redes para tornar suas redes mais otimizadas, seguras e estáveis. Utilizando a experiência adquirida nestes últimos quatro anos, pude identificar algumas características e situações que motivaram o desenvolvimento deste trabalho.

O primeiro fator identificado foi que a maioria das empresas distribuídas não possui seus bancos de dados integrados, ou seja, nestas corporações formadas de várias filiais não existia nenhuma comunicação ou troca de informações entre os bancos de dados das unidades. Esta maneira não integrada é indesejável, pois dificulta ou até inviabiliza a obtenção de uma visão geral da empresa.

O segundo fator foi o crescimento do número de serviços de acesso a Internet de banda larga do tipo dedicado, o que fez com que seu preço tivesse uma grande queda, tornando-se acessível à boa parte das empresas de médio e grande porte.

O terceiro fator foi o crescimento do número de software de banco de dados com suporte parcial ou total a banco de dados distribuídos, o que fez com que seu preço se tornasse mais acessível às empresas.

Porém, o fator principal foi que através do estudo realizado neste trabalho sobre banco de dados distribuído tornou-se possível identificar que poderiam ser utilizadas características desta tecnologia para auxiliar a gerência de redes. Este fator foi o que mais motivou a implementação deste trabalho, pois se pudesse ser feita uma validação da integração das duas tecnologias, tornaria sua utilização uma nova opção para administração de redes.

## 1.1 Objetivos

O objetivo fundamental deste trabalho é validar o processo de integração das tecnologias de banco de dados distribuídos e gerência de redes com a finalidade de tornar as informações de uma empresa distribuída mais integrada, além de tornar o acesso a estes dados mais rápido, fácil e seguro. O enfoque principal será a utilização de características de banco de dados distribuídos para auxiliar a administração de redes.

Pretende-se validar tal estudo através de testes realizados sobre o processo de gerência proposto por este trabalho, além de avaliar os demais processos que internamente utilizam características de banco de dados distribuídos.

São objetivos específicos deste trabalho:

- a) fazer um estudo específico sobre banco de dados distribuído;
- b) desenvolvimento de um estudo sobre o suporte de banco de dados distribuído no ORACLE;
- c) integrar as bases de dados de uma empresa distribuída;
- d) validar o processo de fragmentação horizontal identificando possíveis problemas e tempo de resposta;

e) validar o processo de replicação identificando possíveis problemas e tempo de resposta; e

f) validar o processo de gerência proposto implementando o controle de tráfego descrito neste trabalho.

## 1.2 Trabalhos Correlatos

Este trabalho tem como objetivo fundamental validar a integração das tecnologias de gerência de redes e banco de dados distribuídos com a finalidade de tornar mais fácil e rápida a obtenção de informações de gerência de uma empresa distribuída. Ou seja, para o desenvolvimento deste trabalho foi necessário um estudo detalhado sobre estas duas tecnologias e sua possível integração.

Existem várias correntes de pesquisa que têm como ponto principal de estudo a gerência de redes e suas aplicações, como CARVALHO(1993), que apresenta a infraestrutura e as características principais desta tecnologia.

Outro tópico de boa parte das pesquisas nesta área é a utilização de agentes móveis. Os estudos de RUBINSTEIN(2001), KAHANI(1997) e MAES(1994) apresentam a funcionalidade e algumas de suas aplicações. Outro aspecto importante é que vários estudos têm sido desenvolvidos no sentido de propor e validar o uso de agentes móveis especificamente em aplicações de gerência de redes. A utilização deste paradigma é uma possível alternativa para o modelo proposto neste trabalho.

Também existe um grande material bibliográfico na área de banco de dados distribuídos, como BELL(1992), que apresenta arquitetura, modelos utilizados e características básicas desta tecnologia. Outros pesquisadores mantêm seus estudos focados em aplicações práticas. MURAKAMI(1989) propõe uma arquitetura de controle para redes de comunicações baseadas em banco de dados distribuídos, já BURETTA(1997), faz um estudo específico sobre replicação, identificando ferramentas e técnicas para manipulação de informações distribuídas.



Alguns pesquisadores desenvolveram trabalhos que utilizam as tecnologias de gerência de redes e banco de dados distribuídos como suporte para alcançar seus objetivos. Porém, deve ficar claro que o foco principal destes estudos não era a validação desta integração de paradigmas.

ANEROUSIS(1999) propõe um ambiente de computação distribuído para desenvolver serviços de gerenciamento escalonável, neste estudo são utilizados agentes inteligentes para criar tais serviços no qual as informações de gerência são visualizadas através de aplicações clientes. Neste projeto foi utilizado banco de dados distribuídos para reduzir custos e facilitar a replicação de dados.

Neste estudo, o autor desenvolveu um protótipo em *java* onde foram realizados alguns testes.

HARITSA(1993) propõe o desenvolvimento de uma ferramenta para administração e gerência de redes de comunicação intitulada “MANDATE” (*Managing Networks using Database Technology*), o enfoque principal deste estudo é o desenvolvimento de um sistema de banco de dados que efetivamente suportasse o gerenciamento de redes de grandes corporações.

O presente trabalho apresenta um estudo detalhado da efetiva integração das tecnologias de gerência de redes e banco de dados distribuídos com o objetivo de utilizar algumas características do paradigma de BDD para auxiliar o administrador de redes de uma empresa distribuída. De forma complementar, procura validar tal processo com conceitos, além da implementação de um protótipo onde foram feitos experimentos do qual foram obtidos dados de funcionamento e desempenho. Os experimentos foram realizados utilizando dados reais em um ambiente real.

Seu diferencial principal dos demais trabalhos realizados anteriormente é que nestes estudos a integração era tratada apenas como um suporte para a obtenção de outros objetivos, desta maneira algumas características importantes que poderiam ser utilizadas para facilitar o administrador de redes foram ignoradas. Além disso, nenhum outro trabalho teve como enfoque principal tal tema. Complementarmente, a análise dos experimentos realizados

mostrou que a sua implementação pode ser uma boa opção para auxiliar no gerenciamento de redes.

### 1.3 Organização do Trabalho

Este trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta os principais conceitos sobre banco de dados distribuídos, descrevendo aspectos como características, arquitetura, etc.

O Capítulo 3 apresenta a fundamentação teórica necessária sobre gerência de redes, apresentando conceitos básicos, modelo básico baseado no SNMP, protocolo de gerenciamento, além de características desta tecnologia.

O Capítulo 4 apresenta um estudo sobre banco de dados distribuídos no ORACLE, identificando seu nível de suporte, além de assinalar características e pontos fortes e fracos deste produto.

O Capítulo 5 apresenta toda a parte de validação realizada neste trabalho. Neste capítulo é descrito desde como estava a situação inicial (sistema centralizado) até como foi feita a distribuição dos dados, todo o processo de criação dos *database links*, além de demonstrar como foi implementada a replicação. Por fim, ainda neste capítulo é descrito como foi realizada a validação dos processos mostrando os resultados dos testes realizados.

O Capítulo 6 apresenta as conclusões obtidas a partir do trabalho realizado juntamente com as propostas para trabalhos futuros.

## **2 BANCO DE DADOS DISTRIBUÍDOS**

### **2.1 INTRODUÇÃO**

Nesta seção serão apresentados os conceitos básicos sobre banco de dados distribuídos. Embora nem todos os conceitos sejam totalmente implementados neste trabalho, seu conhecimento facilitará o melhor entendimento das demais seções.

### **2.2 O QUE É BANCO DE DADOS DISTRIBUÍDO?**

Sistema de banco de dados pode ser definido (BELL, 1992) como uma coleção de dados inter-relacionados que se encontram fisicamente distribuídos pelos nós de uma rede de computadores.

**Sistema Gerenciador de Banco de Dados Distribuído (SGBDD)** - é um software que gerencia um banco de dados distribuído de tal modo que os aspectos da distribuição ficam transparentes para o usuário. Assim, o usuário tem a ilusão de integração lógica de dados, sem requerer a integração física de banco de dados. O usuário de um sistema de banco de dados distribuído é incapaz de saber a origem das informações, tendo a impressão de estar acessando um único banco de dados. O SGBDD é responsável pelo gerenciamento global e local, onde cada nó é capaz de processar transações locais, as quais acessam apenas dados daquele único nó, ou pode participar na execução de transações globais, que fazem acesso a dados em diversos nós. A execução de transações globais requer comunicações entre os nós (seja via cabo, fibra ótica etc.). A Figura 1 ilustra um exemplo típico de banco de dados distribuído.

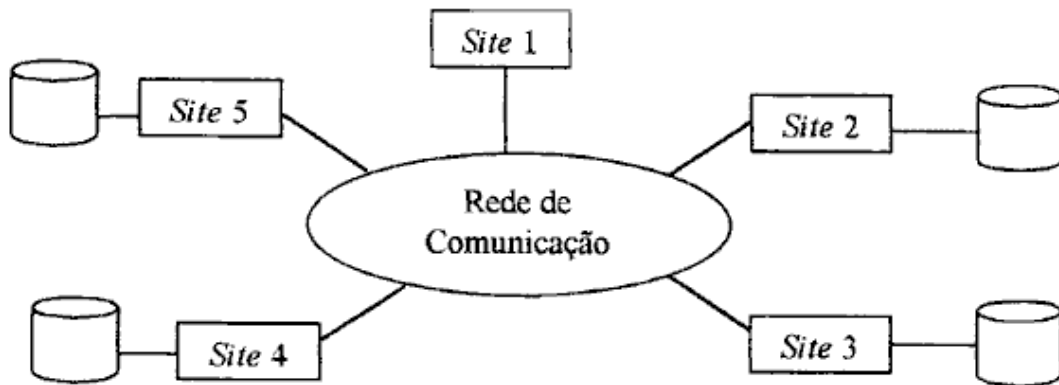


Figura 1 - Banco de dados distribuído

**Sistema Gerenciador de Banco de Dados Centralizado (SGBD)** - é um sistema onde o gerenciamento é feito por um único sistema de computador e todas as requisições são encaminhadas para este *site*. A existência de uma rede de computadores não significa que o sistema é distribuído, pois não existe a distribuição física dos dados. A Figura 2 mostra um exemplo típico de banco de dados centralizado. Note que todas as requisições referentes ao banco de dados são encaminhadas ao *site 2*.

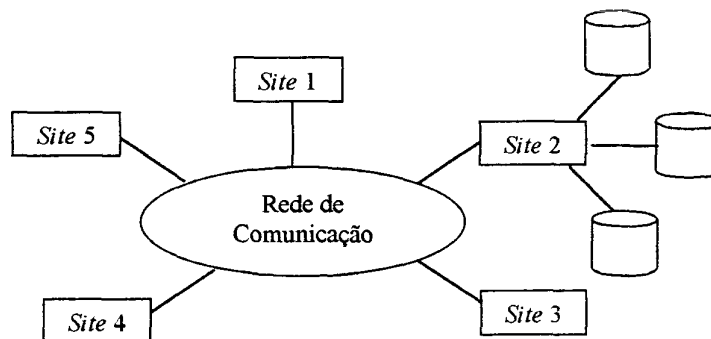


Figura 2 - Banco de dados centralizado em uma rede de computadores

### 2.3 CARACTERÍSTICAS DE UM BANCO DE DADOS DISTRIBUÍDOS

**Autonomia Local** - os sítios, em um sistema distribuído, devem ser autônomos, ou seja, os dados são acessados e gerenciados localmente, as operações locais permanecem no mesmo

sítio e nenhum sítio depende de outro para se tornar operável. Não deve existir dicionário centralizado, controlador de atividades centralizado ou detector de bloqueio mútuo centralizado.

**Independência dos Sítios** - todos os sítios em um sistema distribuído devem possuir o mesmo tratamento. As funções de gerenciamento de dicionário, processamento de pesquisas, controle de concorrência e recuperação não devem depender de um sítio central.

**Operação Contínua** - a entrada de um novo sítio e a instalação de uma nova versão do SGBD são exemplos de operações que não devem alterar em nada o funcionamento do sistema ou das aplicações. Cada sítio é uma unidade funcional independente.

**Independência de Local** - os dados aparecem para os usuários como se estivessem armazenados localmente. A movimentação de dados entre sítios não deve alterar a forma do usuário visualizar os dados. O usuário deve consultar e atualizar dados independente do sítio.

**Independência de Fragmentação** - fragmentação é a divisão de uma relação utilizando-se operações relacionais de restrições e projeções. Esta divisão pode ocorrer de forma horizontal, envolvendo subconjuntos de linhas, de forma vertical, envolvendo subconjuntos de colunas e de formas mistas. A reconstrução da relação original dos fragmentos é feita utilizando-se operadores de junção e de união da linguagem de Manipulação de Dados, *Structres Query Language* (SQL). A visão que um usuário possui de uma relação fragmentada é a mesma que ele possui de uma relação centralizada.

**Independência de Replicação** - um sistema de banco de dados distribuído deve suportar replicação de dados, ou seja, deve permitir que uma relação (ou fragmento) possua uma ou mais cópias em vários locais distintos da rede. As razões básicas de uso de réplicas são a disponibilidade dos dados e o desempenho das aplicações. A localização e manipulação das réplicas de dados devem ser transparentes ao usuário.

**Processamento de Consultas Distribuídas** - a otimização de consultas em um sistema distribuído deve ser também distribuída, envolvendo normalmente uma etapa de otimização global para minimizar o tráfego na rede, seguido das etapas seguintes de otimização local em cada sítio envolvido.

**Gerenciamento de Transações Distribuídas** - dois aspectos importantes no gerenciamento de uma transação distribuída são o controle de concorrência e o controle de recuperação. Em uma transação distribuída, a atomicidade da transação deve ser obtida, da

mesma forma como ocorre em um sistema centralizado. O método geralmente utilizado para manter a integridade dos dados em uma transação distribuída é conhecido como Protocolo Bifásico de Confirmação. Os SGBDs comerciais, em sua maioria, utilizam o bloqueio de linha para atualizações e, para detectar bloqueios mútuos globais, o mecanismo de expiração de tempo.

**Independência de Máquina** - os SGBDDs utilizados em um sistema de banco de dados distribuídos devem suportar qualquer plataforma de hardware, ou seja, o fato de rodar em uma ou outra máquina deve ser transparente ao sistema.

**Independência do Sistema Operacional** - o sistema operacional que executará o SGBDD deve ser transparente ao Sistema de Banco de Dados Distribuído.

**Independência de Rede de Comunicação** - topologia da rede, protocolo, método de acesso, etc, são tópicos transparentes ao sistema distribuído. O sistema de banco de dados distribuído deve rodar em qualquer plataforma de rede.

**Independência de SGBD** - a independência de SGBDs em um sistema homogêneo é muito fácil de ser alcançada. A interface de acesso padrão utilizada é a linguagem SQL, padronizada pela American National Standards Institute (ANSI).

Os problemas se avolumam quando um banco de dados distribuído é implementado em um sistema heterogêneo de SGBDs. Atualmente existem comercialmente SGBDs hierárquicos, redes e relacionais. Rodar transações distribuídas envolvendo três formas distintas de armazenamento e de estruturação dos dados é o grande problema que se apresenta. As soluções de intercomunicação entre SGBDs dão-se através de pontes (*gateways*), utilizando-se de acesso nativo aos dados ou protocolos abertos (ODBC, DRDA etc.).

## 2.4 ARQUITETURA DE UM BANCO DE DADOS DISTRIBUÍDOS

A arquitetura de um banco de dados distribuído é constituída de esquema conceitual global, esquema de fragmentação, esquema de alocação e o projeto físico do banco de dados em cada sítio.

**Esquema Global** - são descritas todas as entidades de dados (relações) e seus relacionamentos. A definição dos dados, em um banco de dados distribuído, é feita da mesma forma que em um banco de dados centralizado.

**Esquema de Fragmentação** - fornece o mapeamento entre relações globais e seus respectivos fragmentos. Determina como relações globais são divididas em fragmentos verticais, horizontais ou mistos.

**Esquema de Alocação** - define em que sítio cada fragmento irá residir fisicamente. Determina de que forma os fragmentos são mapeados para alocações físicas.

## 2.5 PROJETO DE UM BANCO DE DADOS DISTRIBUÍDOS

O Projeto de banco de dados deve reduzir acessos remotos e simplificar os controles dos programas de aplicação, além disso, em alguns casos a utilização de cópias facilita o acesso em caso de indisponibilidade de um determinado sítio e aumenta a confiabilidade do sistema em caso de acidentes na instalação. Outro fator importante é aproveitar a capacidade de cada sítio (armazenamento e processamento) e maximizar o nível de paralelismo de execução da aplicação.

Pode-se afirmar que o projeto deve maximizar o processamento local de cada sítio sem deixar de ser confiável, distribuindo a carga de trabalho sempre que possível e ter um custo acessível.

Existem basicamente duas propostas em projetos de banco de dados distribuídos:

- a) **Top-Down** - utilizado em projetos novos de banco de dados. Geralmente utiliza banco de dados homogêneos. Cria inicialmente o desenho do esquema global, seguido pelo esquema de fragmentação e alocação. Este modelo é complementado com o projeto físico dos dados alocados em cada sítio; e
- b) **Bottom-Up** - utilizado para integração de vários bancos de dados já existentes. Para este caso é necessário um SGBDD heterogêneo, capaz de fornecer interoperabilidade entre os bancos de dados locais.

### 2.5.1 Projeto de fragmentação de dados

Consiste em identificar e agrupar tuplas (fragmentação horizontal) ou atributos (fragmentação vertical) que tenham a mesma propriedade do ponto de vista de sua alocação.

**Fragmentação Horizontal** - é o particionamento das tuplas de uma relação global em subconjuntos disjuntos. A reconstituição da relação global se dá através da união de todos os fragmentos. Determinar a fragmentação horizontal significa determinar as propriedades lógicas dos dados (predicado de fragmentação) e propriedades estatísticas dos dados (números de referências das aplicações para o fragmento).

**Fragmentação Vertical** - é o agrupamento de atributos de uma relação R onde cada atributo de R deve pertencer a pelo menos um fragmento e cada fragmento deve incluir um identificador único de tupla ou chave primária. A relação global pode ser obtida com a operação de junção dos diversos fragmentos. As alternativas para se chegar aos atributos de particionamento são:

- a) **divisão** - consiste na divisão progressiva da relação global em fragmentos;
- b) **agrupamento** - agregam-se progressivamente os atributos para se constituir os fragmentos; e
- c) **fragmentação mista** - é o particionamento de uma relação global R utilizando-se a fragmentação horizontal e vertical.

### 2.5.2 Projeto de replicação de dados

O principal objetivo do uso de réplicas é a melhoria de performance de acesso aos dados. Quando a cópia é somente de leitura recebe o nome de Extrato ou *snapshot*. Quando uma cópia possui a característica de poder ser atualizada é denominada somente de réplica. As cópias podem ser completas ou parciais. As parciais utilizam o log da tabela *master* para efetuar as atualizações.

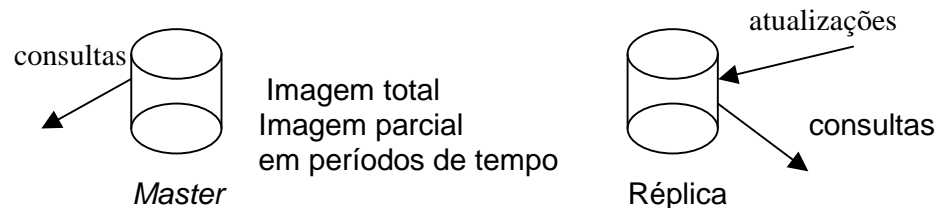
As cópias Extrato podem ser:



- a) **simples** - não possui mecanismo automático de atualização da cópia;
- b) **timestamp** - a renovação dos dados é efetuada a partir de mecanismo de tempo definido; e
- c) **refreshed** - pode ser periódico ou imediato, no primeiro a renovação parcial ou total dos dados é feita em períodos de tempo. Já no segundo, todas as cópias de uma tabela *master* são atualizadas no mesmo instante da atualização da mesma.

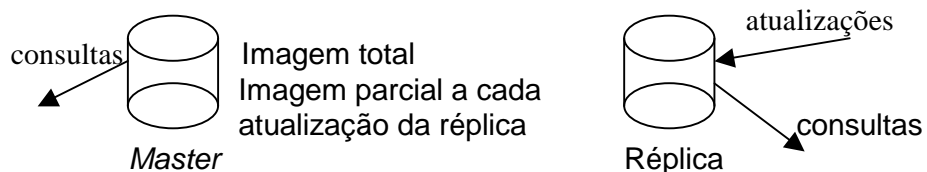
As réplicas podem ser:

- a) **periódicas** - num primeiro momento é feita uma cópia completa ao sítio destino. A partir daí, as réplicas já podem sofrer atualizações. As atualizações efetuadas nas réplicas são periodicamente transferidas para a tabela *master*. A Figura 3 ilustra o funcionamento deste processo; e



**Figura 3 - Réplica periódica**

- b) **contínuas** - depois de realizada uma cópia completa, esta passa a receber atualizações. Cada alteração realizada em uma réplica é imediatamente repassada à tabela *master*. A Figura 4 ilustra o funcionamento deste processo.



**Figura 4 - Réplica contínua**

### 2.5.3 Projeto de alocação de dados

Tem por objetivo definir as alocações físicas dos arquivos do banco de dados distribuído em cada sítio, levando-se em consideração a máxima eficiência nas operações de consulta e atualizações dos dados.

Porém, a definição desta alocação é bem complexa e grande parte das soluções são muito genéricas e complicadas para serem usadas em problemas práticos.

Geralmente a escolha de como serão alocados os dados é feita usando um misto de soluções (algumas até empíricas), chegando a um resultado eficiente (não o ideal) o que torna a consulta de uma transação mais eficaz.

## 2.6 ARQUITETURA DE UM SGBDD

É constituída pelo Diretório de Dados Global (DDG), Gerente de Transações (GT) e Gerente de Dados (GD).

**DDG** - contém a descrição do banco de dados distribuído.

**GT** - responsável pelo controle dos acessos ao BDD.

**GD** - é a interface com o SGBD local, tendo como principal função as traduções entre banco de dados heterogêneos.

Para cada sítio existirá um SGBD local, além de uma cópia do SGBD global. Cada nó poderá ou não conter *paste* do diretório global. Isto dependerá da estratégia de alocação adotada.

### 2.6.1 Gerência de transações

Uma transação é definida como uma série de ações, iniciada por um simples usuário/programa de aplicação, que deve ser executada de forma atômica, ou seja, efetivação de todas as ações ou de nenhuma das ações.

### **Propriedades:**

- 1) **atomicidade** - uma transação é uma unidade indivisível;
- 2) **consistência** - uma transação não altera a consistência de um banco de dados;
- 3) **independência** - uma transação executa independentemente de qualquer outra. Estados parciais de uma transação incompleta não são visíveis a outras transações; e
- 4) **durabilidade ou persistência** - as ações realizadas pela transação, após o seu término, caso confirmadas, são permanentemente preservadas no banco de dados independente de falhas posteriores.

A gerência de transações tem como objetivo garantir atomicidade, durabilidade, seriabilidade e propriedades de isolamento. Deve também obter menores custos em termos de memória principal, *Central Processing Unit* (CPU) e número de mensagens de controle transmitidas. Além disso, deve maximizar a disponibilidade.

#### **2.6.2 Controle de concorrência**

Quem faz o controle de concorrência é chamado de Escalonador, que tem como principal função evitar interferências entre transações. A comunicação entre o programa de aplicação e o Escalonador é realizada pelo Gerente de Transações (coordena operações de banco de dados na execução de aplicações). Qualquer técnica de controle de concorrência adotada pelo sistema deve garantir que toda execução concorrente de um conjunto de transações seja serializável. Isto equivale a dizer que, em um conjunto de transações, cada transação deve ser executada completamente antes que a próxima transação, envolvendo os mesmos dados, inicie.

#### **2.6.3 Gerência de recuperação**

Sua principal função é de em caso de falha do sistema, identificar quais transações têm que ser desfeitas e quais têm que ser refeitas e, em seguida, efetuar as necessárias alterações no banco de dados.

#### **2.6.4 Integridade**

A integridade de um banco de dados está relacionada à consistência, exatidão, validade e precisão dos seus dados. Uma violação de integridade pode ocorrer por vários motivos, os mais comuns são: inserção de uma chave inválida em uma tabela, falha no algoritmo de controle de concorrência e falha na gerência de recuperação.

A solução de conflitos de regras de integridade local e global depende do nível de integração entre os vários sítios participantes em um banco de dados distribuído. Se existe um forte controle global, então as regras de integridade globais devem ter prioridades sobre as locais. Por outro lado, se a federação dos bancos de dados distribuídos não possuir praticamente nenhum controle global, as regras de integridade local devem prevalecer.

Antes de se definir regras de integridade globais em um banco de dados distribuído, deve-se verificar as regras de integridade locais para que não ocorram conflitos.

#### **2.6.5 Segurança**

Cada SGBD local possui a responsabilidade da segurança de seus dados. A violação de segurança ocorre quando deliberadamente há uma tentativa de acesso não autorizado a determinados dados.

**Identificação e Autenticação** - para se ter acesso aos dados de um determinado banco de dados, o usuário/programa de aplicação deve fornecer sua identificação seguida de sua senha. Para conexões remotas, pode-se usar o usuário ativo da presente sessão, ou informando-se usuário/senha.

**Distribuição de Regras de Autorização de Acesso** - em um ambiente de banco de dados distribuído, as regras de autorização de acesso aos dados definidos para um sítio devem ser replicadas aos outros sítios participantes.

**Encriptação** - a encriptação dos dados é uma forma de coibir acessos não autorizados. É também uma forma de resolver o problema de pessoas que burlam as regras de segurança

do SGBD ou que tenham acesso aos *frames* do protocolo da rede de comunicação. A encriptação dos dados pode ser realizada através de vários métodos, entre eles os métodos *Data Encryption Standart* (DES) ou *Public Key Cryptosystems* (PKC).

**Visão Global** - através de definições de visões, os dados são disponibilizados de uma forma restritiva, ou seja, limita-se o número de colunas acessadas como também o número de linhas em uma tabela. Através dos comandos *Grant* e *Revoke* poder-se-á limitar o acesso ou liberá-lo.

A definição das regras de segurança é geralmente delegada ao Administrador de Banco de Dados (DBA).

### **2.6.6 Administração de bancos de dados distribuídos**

Em um sistema de bancos de dados distribuído, o DBA exerce atividades tais como: Otimização de alocação de fragmentos e replicação dos dados; Monitora a performance do sistema e efetua correções; Migra dados de um sítio para outro, entre outras. O Administrador de Banco de Dados (DBA) pode ser considerado como uma pessoa, ou grupo de pessoas, responsável pelo gerenciamento dos recursos dos dados dentro de uma organização.

Além disso, o DBA tem o papel de gerente de segurança, definindo para isso regras aos usuários do banco de dados (utilizando visões, *Grants*, etc), podendo ainda determinar o nível máximo dos recursos, tais como: tempo de CPU, VOs lógicos em disco, tempo de conexão, etc. Se um usuário excede tais limites, sua sessão ou requisição, é encerrada imediatamente.

## **3 GERÊNCIA DE REDES**

### **3.1 INTRODUÇÃO**

Nesta seção serão apresentados os conceitos básicos sobre gerência de redes. Embora nem todos os conceitos sejam totalmente implementados neste trabalho, seu conhecimento facilitará o melhor entendimento das demais seções.

### **3.2 O QUE É GERÊNCIA DE REDES?**

Redes de computadores, como qualquer outro sistema, precisam ser gerenciadas para que possam prover um serviço com qualidade. A qualidade envolve a disponibilidade, o desempenho, etc. Podemos observar neste momento que não é apenas a rede de computadores que precisa de gerência. Na realidade, o usuário está mais interessado nos serviços oferecidos pela rede do que na rede em si. Quando se fala de gerência, portanto, deve-se incluir não somente a infra-estrutura de comunicação, mas também as aplicações que provêm os serviços finais aos usuários.

A solução mais usual para o gerenciamento de rede consiste em utilizar um computador que venha a interagir com os diversos componentes para deles extrair as informações necessárias para o seu gerenciamento. Esta solução leva à criação de um banco de dados no computador gerente, contendo informações adequadas e necessárias para apoiar o diagnóstico e busca de soluções para problemas. Isto envolve um esforço para rastrear, identificar e resolver situações de falha física, mau funcionamento de softwares ou aspectos de treinamento que afetam o funcionamento adequado da rede.

### 3.3 ÁREAS DA GERÊNCIA DE REDES

**A Gerência de Falhas:** deseja-se aqui manter a rede funcionando. Isto implica em identificar os elementos da rede que falharam (equipamentos, canais de comunicação, processos de aplicação, etc.) e responder a tais eventos de forma a consertar as falhas e restabelecer o funcionamento da rede. É também importante que se mantenha um histórico das falhas para, por exemplo, substituir equipamentos com altas taxas de falha.

**A Gerência de Configuração:** redes de computadores utilizam equipamentos complexos (roteadores, por exemplo). Esses equipamentos devem ser configurados corretamente de forma a desempenharem suas funções adequadamente. Uma das funções da gerência é de facilitar esta configuração. Isso pode envolver questões de endereçamento, tabelas de roteamento, além de dezenas de outros fatores.

**A Gerência de Contabilização:** é de vital importância para que os custos e volumes de recursos utilizados pelos usuários sejam identificados e registrados de forma correta. A contabilização abrange, praticamente, todas as camadas do modelo *Open Systems Interconnection* (OSI), pois todos os recursos utilizados são passíveis de monitoração.

**A Gerência de Desempenho:** é importante para garantir a qualidade de serviço acordada com os usuários que seja assegurado que esta é atingida com os menores custos possíveis. Pode-se, por meio do gerenciamento de desempenho, adequar os meios de comunicação utilizados pelos usuários para as suas reais necessidades. O gerenciamento de desempenho está diretamente relacionado ao planejamento da capacidade do sistema sob gerenciamento.

**A Gerência de Segurança:** diz respeito ao uso do gerenciamento de redes para monitorar e controlar os mecanismos de segurança. Estes mecanismos envolvem desde o controle de acesso aos sistemas computacionais até o acesso às informações sigilosas que trafegam nos circuitos de dados. É difícil, na prática, distinguir os mecanismos de segurança do gerenciamento de segurança. Um sistema só pode ser seguro se os protocolos que o gerenciam não estiverem comprometidos e a única forma de assegurar o não comprometimento é torná-los seguros.

### 3.4 MODELO DE GERÊNCIA BASEADO NO SNMP

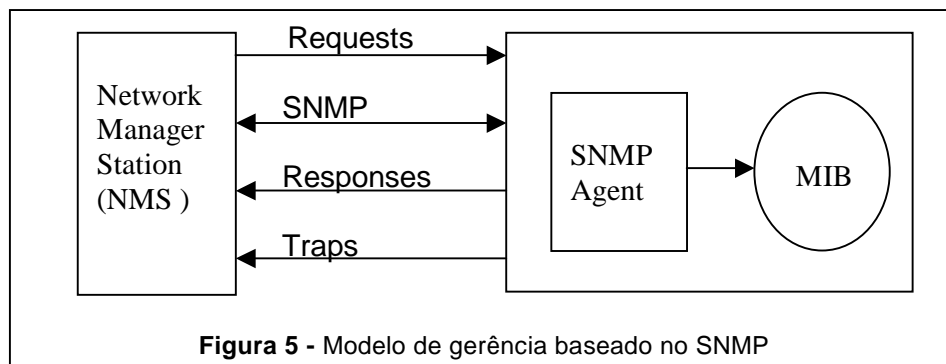
O modelo básico baseado no *Simple Network Management Protocol* (SNMP) é formado de no mínimo uma estação de gerenciamento de rede *Network Management Station* (NMS), vários nós gerenciados e um protocolo de gerenciamento para comunicação da informação de gerenciamento entre uma NMS e um nó gerenciado. Uma NMS é um *host* que executa aplicações de gerenciamento de rede e para isso, se utiliza do protocolo de gerenciamento. Um nó gerenciado poderia ser um *host*, um *gateway* ou meio de transmissão. Esse modelo objetiva ter uma funcionalidade mínima na maioria dos nós e uma maior funcionalidade numa pequena fração dos nós na rede.

Cada nó gerenciado suporta os protocolos operacionais, o protocolo de gerenciamento e a entidade de interface. A entidade de interface projeta as variáveis "reais" definidas e usadas pelos protocolos operacionais como objetos gerenciados. Essa entidade também define um conjunto de operações nessas variáveis. O conjunto de objetos e operações gerenciados são armazenados na *Management Information Base* (MIB). Cada NMS executa o papel de um gerente e suporta o protocolo e aplicações de gerenciamento. Os objetos definidos neste esquema podem ser dois tipos: Escalares e Tabela. Os objetos do tipo Escalar funcionam como variáveis e os objetos Tabela são listas de variáveis.

O SNMP se baseia no paradigma "*request-response*". Os papéis de gerente e agente são fixos. O gerente, pela mudança de um valor de um objeto gerenciado, pode instruir o agente a executar uma determinada operação de gerenciamento. O gerente pode requerer os valores de certos objetos gerenciados (que são modificados pelo nó gerenciado) para determinar o estado atual do nó.

Todo aspecto de comunicação desse modelo está reduzido a apenas modificar e requerer valores (*SET* e *GET*, respectivamente) dos objetos gerenciados. Ele define objetos que têm as seguintes características: Acesso, Estado, Sintaxe e Nome. O Acesso controla os privilégios *readwrite*; Estado classifica os objetos gerenciados como opcionais, obrigatórios e obsoletos, na implementação, o nome denota o nome do objeto gerenciado; A Sintaxe define a especificação da sintaxe do objeto; O Nome usado para identificar um objeto é chamado *Object Identifier* (*OID*) que é um nome único na árvore de nomeação global. A Figura 5 ilustra o funcionamento deste modelo.





O protocolo SNMP usa como suporte o *User Datagram Protocol* (UDP) e suporta as seguintes operações básicas:

- a) **get-request** - permite ao gerente requerer de um agente o valor de uma variável particular;
- b) **get-nextrequest** - permite ao gerente MIB transversal, ou seja, se o agente conhece qual o ramo da árvore da MIB onde está a variável, mas não especificamente o nome da variável, então ele pode emitir uma operação *get-nextrequest*;
- c) **set-request** - usada pelo gerente para modificar ou não o valor de uma determinada variável; e
- d) **trap** - permite ao agente assincronamente relatar a ocorrência de um evento ou falha para o gerente.

### 3.5 FORMATO DAS INFORMAÇÕES GERENCIÁVEIS

Dentro das bases de dados gerenciáveis as informações encontram-se agrupadas em objetos gerenciáveis que contém: Nome, Atributos, Conjunto e Operações (métodos). Os objetos gerenciáveis são recursos reais que estão sendo gerenciados pelo ambiente SNMP.

Os objetos SNMP têm por propriedades:

- a) **nome único** - chamado de Identificador de Objeto (*OBJECT IDENTIFIER*);

- b) **atributos** - com tipos de dados, descrição que incluem detalhes requeridos para sua implementação e informações de *status*, que informam se esta definição de objeto é atual ou obsoleta; e
- c) **operações válidas** - tais como leitura e escrita, que podem ser realizadas sobre o objeto.

### 3.6 NOMEAÇÃO DE VARIÁVEIS SNMP

Os nomes das variáveis SNMP são hierárquicos, não-tipados e baseados no tipo *Application Programming Interface* (ASN.1) *object identifier*. Cada variável usa a sintaxe ASN.1 da forma {x,y}, onde x e y são seqüências de componentes ASN.1 *object identifier*. A primeira parte x é associada e fixada quando o tipo da variável é definido. A segunda parte y é associada e fixada quando a variável é instanciada na MIB.

Os nomes das variáveis SNMP são únicos somente no sistema. O mesmo nome de variável é reutilizado para instanciar variáveis em diferentes sistemas. Para fornecer nomeação global, os nomes das variáveis devem ser qualificados como nomes globalmente únicos no sistema. Entretanto, o protocolo SNMP não faz previsão para transportar essa informação adicional. Uma limitação que isso impõe é que um gerente SNMP não tem definido como dirigir uma requisição para outro gerente SNMP para executar uma operação numa variável MIB num agente. A Figura 6 ilustra esta estrutura de nomes.

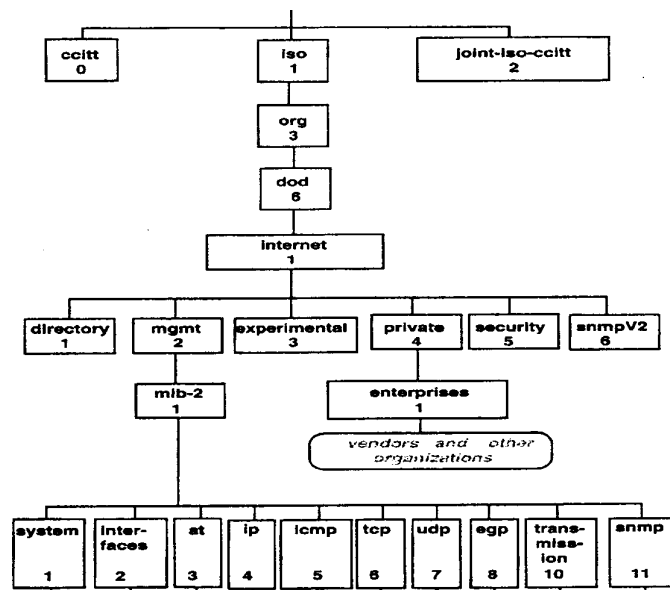


Figura 6 - Árvore de identificadores de objetos

A estrutura do nome para uma variável SNMP é fixada quando a variável é projetada. O único meio para um usuário modificar essa estrutura é definir um novo tipo de variável com um nome único.

Por exemplo, para identificar uma instância da variável *system*, a classe de objetos para *system* é:

**Iso org dod internet mgmt mib system**

1 3 6 1 2 1 1

Assim, o nome da instância que é referenciado por {x.y}, sendo y=0, por ser a primeira instância seria 1.3.6.1.2.1.1.0.

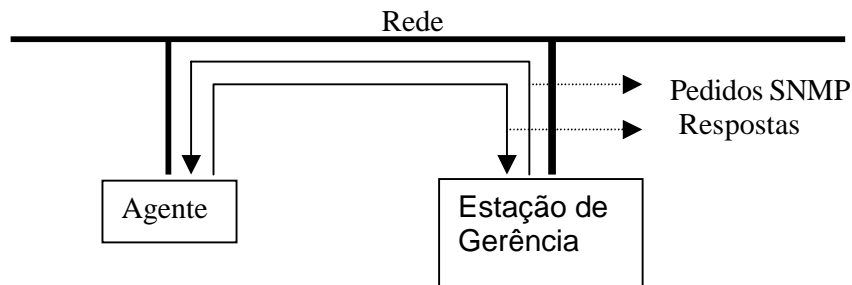
### 3.7 PROTOCOLO DE GERENCIAMENTO

O protocolo SNMP segue o paradigma da depuração remota, ou seja, cada nó gerenciável é visto como tendo diversas variáveis. A monitoração do nó pode ser realizada lendo-se as suas variáveis e o controle pode ser efetuado alterando-se os seus valores.

O SNMP é um protocolo assíncrono, com envio de pedidos e obtenção de respostas, sendo baseado num enfoque de *pooling* direcionada para *trap*. Quando o evento ocorre, o nó gerenciável envia um *trap* muito simples para a estação de gerenciamento. Desta forma, esta estação de gerenciamento é avisada e fica responsável a tomar as medidas adequadas, entre elas, inicializar as operações com o nó gerenciado para determinar a natureza do evento.

### 3.8 FLUXO DE OPERAÇÕES

Genericamente, uma interação SNMP consiste em um pedido seguido de uma resposta, como representado na Figura 7.



**Figura 7-** Interação de uma requisição SNMP

Onde nesta operação SNMP, quatro resultados são possíveis:

- a) resposta sem exceção ou erro;
- b) resposta com uma ou mais exceções;
- c) resposta com um erro; e
- d) *timeout* ou demorou demais para responder.

Diferente da notificação, ou *trap*, que gera uma simples *Protocol Data Unit* (PDU) que é repassada para o gerente.

### 3.9 COMUNICAÇÃO NO PROTOCOLO SNMP

O protocolo SNMP opera sobre um serviço sem conexão fornecido pelo UDP e o *Internet Protocol* (IP).

Como um serviço sem conexão IP e, por extensão UDP, não fornece nenhuma indicação, nem para o remetente nem para o destinatário, os pacotes podem ser perdidos ou descartados por uma série de razões, tais como:

- a) destinatário não disponível;

- b) pacote muito grande a ser transmitido pelo sistema intermediário;
- c) congestionamento da rede; e
- d) perda de conectividade.

O modelo sem conexão não fornece nenhuma maneira para controlar o fluxo de dados entre um gerente e um agente. Os gerentes que não são habilitados para processar grandes volumes de *traps*, não tem outra opção, senão descartá-los. O protocolo SNMP parcialmente resolveu esse problema usando os *traps* ponderadamente, para que um gerente saiba que algum evento ocorreu. Ele está acima do gerente para requerer mais informação sobre o evento com operações *GET* e *GET-NEXT*. O modelo sem conexão não fornece a um gerente a oportunidade para trocar e negociar informações tais como capacidades, parâmetros de segurança e outros parâmetros operacionais.

Já que o modelo SNMP permanece com *polling*, o gerenciamento baseado em eventos não é possível. As aplicações que necessitam de informações periódicas, tais como aquelas para gerenciamento de desempenho, devem periodicamente "convidar" cada agente para coletar tais informações.

O protocolo SNMP adapta-se facilmente para operar sobre muitos outros protocolos, incluindo protocolos orientados a conexão, sendo esta uma das suas grandes vantagens.

### 3.10 FERRAMENTAS AUTOMATIZADAS PARA GERÊNCIA

Em função do aumento da complexidade, decorrente principalmente do aumento do número de pontos e da inclusão de novas tecnologias, o gerenciamento manual tornou-se praticamente inviável. Os administradores de redes necessitavam de ferramentas que facilitassem a tarefa do gerenciamento. Desta necessidade surgiram diversas ferramentas automatizadas que facilitaram tais tarefas.

A maioria das ferramentas automatizadas existentes atualmente foram desenvolvidas utilizando o protocolo SNMP. Porém, a maioria delas não possibilita o gerenciamento completo da rede. Buscando isso a ISO desenvolveu o modelo de gerenciamento OSI que utiliza o protocolo *Common Management Information Protocol* (CMIP). A tendência é a adoção do modelo OSI como padrão na área de gerenciamento de redes. Entretanto, o SNMP

continua tendo grande aceitação comercial devido a grande variedade de softwares de gerência e a facilidade de uso e implantação.

## 4 BANCO DE DADOS DISTRIBUÍDOS EM ORACLE

### 4.1 INTRODUÇÃO

Nesta seção será apresentado o funcionamento do banco de dados Oracle em ambiente distribuído, além de mostrar as características deste produto, identificando seus benefícios e dificuldades.

### 4.2 ORACLE COMO SGBDD

O SGBD Oracle possui diversas características de um SGBDD. Para a discussão destas características, algumas definições relativas à arquitetura do produto devem ser entendidas.

Quando dados são requisitados de um banco de dados Oracle remoto para um servidor de banco de dados local, este se comunica com o banco de dados através da rede de comunicação que interliga os dois equipamentos. A comunicação é gerenciada pelo produto Oracle SQL\*Net, que provê uma interface para conectar clientes e servidores que operam em diferentes computadores da rede. A arquitetura Oracle permite a interligação de bancos de diferentes versões do SGBD Oracle, além de banco de dados de outros fornecedores, para criar um Sistema de Banco de Dados Heterogêneos e Distribuídos. Quando conexões de um *site* Oracle para um *site* remoto (Oracle ou não) são inicialmente estabelecidas, o *site* Oracle registra as características do SQL, entre outras informações, de cada sistema remoto e associa *gateways*. Contudo, se o SGBD do *site* remoto não suportar algum comando específico do SQL Oracle, ele será eliminado ao passar pelo *gateway*.

Em um SBDD, cada banco de dados deve ter um único nome global. Nomes globais de banco de dados identificam cada banco de dados na rede. Na arquitetura Oracle, um nome de banco de dados global é composto de dois componentes: Nome do Banco de Dados, com no máximo oito caracteres, e o Domínio da Rede que contém o banco de dados.

O SGBD Oracle garante que cada banco de dados tem um conjunto único de esquemas e, em cada esquema, um nome de objeto (*e.g.* tabela, visão, etc) é único. Assim, se adicionamos o nome global do banco de dados remoto ao nome do objeto remoto, qualquer elemento (objeto ou coluna) pode ser referenciado em um banco de dados local. *Aliases* também podem ser criados localmente para obter transparência de localização. Em cada *site*, *aliases*, nomes de objetos e outras informações sobre o esquema são armazenados no catálogo do banco de dados local.

O SBDD Oracle implementa o conceito de interconexão entre dois ou mais bancos de dados Oracle através de ligações chamadas *database links*. Cada *database link* define um caminho (*path*) para um banco de dados remoto. São usados quando um comando SQL referencia um objeto global. Ao processar o comando, o SGBD Oracle tenta localizar o *database link* que identifica o nome do banco de dados especificado na referência ao objeto global. O objeto global é identificado quando sua referência é precedida por “@”. Quando a ligação é encontrada, o SGBD Oracle utiliza essa ligação para estabelecer uma conexão com o banco de dados remoto e completa a execução da instrução SQL.

#### 4.3 TIPOS DE *DATABASE LINKS* USADOS NO ORACLE

Existem vários tipos de *database links*:

- a) **privado** - criado para um usuário específico. Só pode ser usado pelo próprio dono (*owner*) da ligação quando submete uma instrução SQL referenciando um objeto global. Outros usuários podem utilizar ligações deste tipo se existirem sinônimos (*aliases*) ou autorizações definidas através de instruções *GRANT*;



- b) **pública** - criado para o grupo especial de usuários chamado *public* (público). Um *database link* deste tipo pode ser usado por qualquer usuário do banco de dados associado;  
e
- c) **rede** - criado ou gerenciado por um serviço de domínio de rede. Pode ser usado por qualquer usuário de qualquer banco de dados da rede. Pressupõe a existência de um servidor de nomes (diretório).

#### 4.4 CARACTERÍSTICAS DO ORACLE

Como já mencionado, o SBDD Oracle implementa a maioria das características de distribuição de dados discutidas na seção 2.3, tais como:

- a) **a autonomia local e transparência** - estão presentes no SGBD Oracle. O administrador de banco de dados (DBA) é capaz de gerenciar, criar *links*, atualizar réplicas e definir critérios de segurança às suas bases locais sem interferir nas aplicações que as acessam;
- b) **a integridade** - o padrão ANSI/ISO de restrição de integridade Referencial e de Entidade é completamente implementado pelo Oracle. Assim a integridade é garantida pelo SBDD através de controles internos, onde uma determinada transação é repassada através da rede para os bancos envolvidos;
- c) **disponibilidade** - para aumentar a disponibilidade, réplicas das tabelas podem ser criadas, oferecendo maior velocidade de acesso aos usuários locais;
- d) **expansibilidade** - é possível, pois o administrador local tem autonomia para definir novos *links* com outros *sites* e novas replicações em suas bases;
- e) **controle de Concorrência** - o Oracle emprega bloqueio a nível de linha para os dados e índices, propiciando assim um nível maior de concorrência. No SGBD Oracle, a consulta não bloqueia a atualização e o contrário também é verdadeiro, ou seja, a atualização de um determinado dado não evita o acesso concorrente ao mesmo. O Oracle possibilita consulta consistente, isto é, todos os dados de uma consulta são uma imagem do momento em que se iniciou a execução da mesma. Todas as atualizações

(mesmo confirmadas) sofridas após ter sido iniciada uma consulta não são disponibilizadas a esta consulta.

O Oracle não emprega o bloqueio escalonável (linha, tabela, *tablespace*), ou seja, o bloqueio é sempre a nível de linha, a menos que a transação informe explicitamente o bloqueio em toda tabela;

- f) ***otimização de consulta baseada em custos*** - o otimizador do Oracle baseia-se em contabilizações estatísticas, tais como número de linhas da tabela, índices definidos, tamanhos médios de campos chaves, etc, para determinar o caminho de acesso mais eficiente para uma determinada consulta.

Periodicamente deve-se executar um utilitário que colete todas as informações estatísticas das tabelas dos bancos de dados, gravando-as nas tabelas do catálogo do SGBD. Ao se efetuar uma consulta, o otimizador irá acessar as informações estatísticas contidas no catálogo para criar o plano de acesso mais eficiente aos dados;

- g) ***limitador de recursos*** - com o Limitador de Recursos do Oracle, o DBA consegue controlar os recursos do sistema que um determinado usuário pode possuir.

O DBA pode determinar o nível máximo dos recursos, tais como: tempo de CPU, VOs lógicos em disco, tempo de conexão, etc. Se um usuário excede tais limites, sua sessão, ou requisição, é encerrada imediatamente;

- h) ***processamento de consulta distribuída*** - um simples comando SQL pode consultar dados de múltiplos Bancos de Dados e fazer de forma transparente consultas complexas.

O usuário nunca necessita saber onde o dado está localizado fisicamente e as aplicações nunca têm que ser recodificadas se um dado migra de um nó para outro;

- i) ***gerenciamento de transações distribuídas*** - o Oracle emprega um mecanismo transparente de Confirmação em Duas Fases (*Two-Phase Commit*) para implementar integridade de atualizações de transações distribuídas.

Transações que atualizam múltiplos nós são confirmadas (*Committed*) com o comando padrão SQL (*commit*). Nenhuma codificação se faz necessária. O Oracle detecta e resolve todas as falhas automaticamente, fazendo com que uma transação distribuída seja totalmente desfeita (*Rollbacked*) ou confirmada (*Committed*).

O Oracle *Two-Phase Commit* pode também interoperar com outros Gerenciadores de Banco de Dados como, por exemplo, o DB2; e

- j) **Integração com Outros SGBDs** - o Oracle consegue conversar com um grande número de outras fontes de dados através de seus produtos de conectividade SQL\*CONNECT e *Transparent Gateway*.

#### 4.5 DISTRIBUIÇÃO DE DADOS NO ORACLE

Na definição de um banco de dados distribuído, pode-se utilizar Visões, Sinônimos, Procedimentos, Funções, Gatilhos e *Snapshots*.

**Visões** - podem ser usadas para providenciar transparência de localização para tabelas locais e remotas em um sistema de banco de dados distribuído.

Quando acessa uma visão, o usuário não sabe, ou não necessita saber, onde o dado está fisicamente gravado, ou se dados de uma ou mais tabelas estão sendo acessados.

**Sinônimos** - são muito úteis em ambientes distribuídos e não distribuídos, porque eles escondem a identidade do objeto, por exemplo, a localização deste. Se o nome de um objeto deve ser trocado ou movido, apenas o sinônimo deve ser redefinido; as aplicações baseadas em sinônimos continuam funcionando sem modificações.

**Procedimentos Armazenados** - a transparência de localização também pode ser efetivada através de unidades de programas PL/SQL contendo comandos SQL que referenciam dados remotos.

**Gatilhos** - o Oracle permite que se definam procedimentos que são implicitamente executados quando comandos SQL *Insert*, *Update* e *Delete* são usados em uma tabela. É similar a um Procedimento Armazenado, podendo conter comandos SQL e PL/SQL.

Na definição, pode-se especificar que o gatilho dispare para cada linha envolvida no comando SQL ou somente uma vez por comando. É também informado se o gatilho dispara antes ou após o comando envolvido.

O SBDD Oracle implementa na sua totalidade a estratégia de distribuição do tipo replicação de dados através de dois conceitos:

a) **Snapshots** - é uma forma de replicação de uma determinada tabela, onde tem-se uma tabela *master* e réplicas distribuídas por vários *sites*. Todas as alterações feitas nas réplicas são efetivadas na tabela *master*. *Snapshots* podem ser somente para leitura ou gravação. Nos acessos unicamente para leitura, o usuário apenas consulta os dados. Nesta condição, o acesso é mais rápido, uma vez que o Oracle não necessita montar uma estrutura de controle sobre os dados. Por sua vez, os acessos para gravação permitem atualização das réplicas, e posterior atualização da tabela *master*; e

b) **Multi-Master** - é uma forma de replicação de um determinado esquema. Definições de um usuário (tais como tabelas, índices, gatilhos, seqüências, etc) podem ser replicadas por vários *sites* e sua alteração em um determinado *site* é posteriormente sincronizada nos outros. Nesta forma de replicação, *snapshots* podem ser ligados para cada *site master*.

A diferença dos dois conceitos se resume à disponibilidade da tabela na rede. Em um *snapshot*, caso o *site* onde reside a tabela *master* fique inativo, as réplicas não serão sincronizadas.

As alterações em uma determinada réplica não serão efetivadas em outras réplicas até que a tabela *master* esteja ativa. Isto não acontece com uma replicação *Multi-Master* onde todas as alterações de cada réplica são sincronizadas com as outras, ficando somente o *site* com problemas em estado inativo.

Não importando a estratégia de replicação utilizada, as sincronizações das réplicas podem ser efetivadas de duas formas:

a) **síncrona** - as informações são replicadas imediatamente para o *site master* (*snapshot*) ou para outras réplicas (*Multi-master*); e

- b) *assíncrona* - as informações são replicadas sob critério de intervalo de tempo definido pelo administrador.

Finalmente, a estratégia de distribuição por fragmentação não é totalmente implementada no SGBD Oracle. Na verdade, expõe alternativas como a replicação de partes dos objetos através de *subqueries*, onde podemos definir as linhas e colunas que cada *site* conterà. Contudo, sempre existirá pelo menos uma tabela que conterà todas as linhas e colunas. Outra alternativa é definição de tabelas em vários *sites*, onde o acesso entre eles se dará através de *database links*. Porém, cada aplicação deve decidir qual *database link* acessar.

## **5 IMPLEMENTAÇÃO E DESENVOLVIMENTO DA INTEGRAÇÃO DE BDD E GERÊNCIA DE REDES**

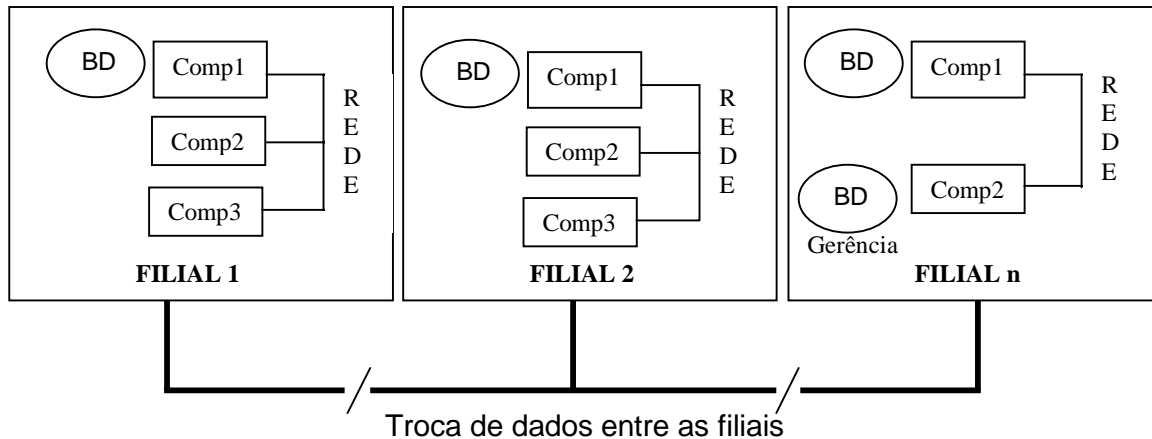
### **5.1 INTRODUÇÃO**

Geralmente a gerência de redes de empresas formadas de várias filiais que estão geograficamente distantes e que possuem redes heterogêneas é feita ou de forma individual (em cada filial), o que dificulta a análise da empresa como um todo, ou utiliza técnicas de agentes móveis, que em alguns casos é difícil de ser colocada em prática (por fatores técnicos e financeiros) e que possui algumas restrições. Nestas empresas não existe nenhuma comunicação ou troca de dados entre as filiais, apesar de que geralmente estas unidades usam produtos que poderiam ser integrados. A utilização de técnicas de banco de dados distribuídos nesta situação tornaria os dados da empresa mais integrados, ou seja, qualquer informação de qualquer filial poderia ser consultada de forma rápida, fácil e segura.

A integração de banco de dados distribuídos e gerência de redes pode ser uma boa opção para médias e grandes empresas que já possuam um banco de dados com suporte a tecnologia de SBDD (o que ocorre na maioria das empresas deste porte) ou para novos projetos que desejam integrar as informações de um empreendimento formado de várias unidades distribuídas.

Neste trabalho será implementado um protótipo desta integração de tecnologias com o objetivo de validar este processo. Os próximos capítulos descrevem como integrar as informações de uma empresa que estão dispersas em vários bancos de dados a fim de facilitar e agilizar sua disponibilização.

## 5.2 SITUAÇÃO ATUAL



**Figura 8** - Situação atual

Nesta seção será demonstrado o ambiente de uma empresa genérica formada de “n” filiais, onde cada filial possui uma rede de computadores distinta e um banco de dados independente. Este ambiente é muito utilizado comercialmente, por este motivo é o ponto de partida do protótipo que será desenvolvido neste trabalho a fim de mostrar os benefícios e dificuldades de se integrar banco de dados distribuídos e gerência de redes.

Neste ambiente inicial cada filial possui um banco de dados individual no qual são armazenadas todas as informações sobre ela (Dados Cadastrais, Financeiros, Comerciais, etc).

Geralmente o banco de dados utilizado em todas as filiais é do mesmo fabricante (Homogêneos), o que facilita a comunicação entre eles. Além disso, geralmente estes bancos de dados têm suporte total ou parcial à técnicas de BDD (Oracle, DB2, etc).

Nem todas as empresas usam técnicas de gerência de redes, nas que utilizam, os dados obtidos de cada filial ou não são armazenados (são utilizados apenas para consulta imediata e depois descartados) ou são armazenados em um banco de dados independente (BD proprietário de alguma ferramenta de gerência), ou seja, nem todos os dados de cada filial podem ser acessados diretamente de um mesmo banco de dados.

Através da Figura 8, pode-se notar que não existe nenhuma conexão direta entre os bancos de dados das unidades da empresa, ou seja, não existe nenhuma troca de informações

entre estas unidades. Essa maneira não integrada de guardar o dados dificulta a obtenção de uma situação geral, isso em todas as áreas (Comercial, Financeira etc.). Assim, se quisermos saber quais os produtos que vendem mais, quais as filiais que produzem mais, teríamos que tirar vários relatórios em cada filial para poder ter uma visão geral e tirar conclusões, isso torna este processo lento e pouco prático.

Na área de gerência de redes ocorre uma situação parecida, ou seja, as soluções de gerência são encontradas utilizando como banco de informações os dados da filial em questão e não da empresa como um todo. Isso dificulta a definição de regras gerais (já que não existe um histórico de problemas e soluções utilizadas para toda a empresa), além de gerar custos desnecessários proveniente do tempo perdido e soluções errôneas utilizadas que podiam ser evitadas caso os dados fossem integrados.

Fica claro que a situação atual possui vários problemas e está longe de ser ideal. Na próxima seção será proposta uma possível solução para o que foi considerado como o problema mais grave desta situação, que é a não integração dos dados.

### 5.3 PROPOSTA DE INTEGRAÇÃO DE BANCOS DE DADOS

Para que se possa integrar os bancos de dados do ambiente descrito na seção anterior, deve-se utilizar os seguintes passos:

O primeiro passo seria a integração dos dados dentro da filial, ou seja, as informações de gerência que estão armazenadas num banco de dados independente devem ser incorporadas ao banco de dados geral. Este passo é relativamente fácil, pois se trata apenas de uma cópia de informações de um banco de dados para outro.

O segundo passo seria a integração dos bancos de dados das filiais, ou seja, utilizar alguma solução que possibilite a consulta e/ou alteração dos dados de todas as filiais de forma transparente, fácil e rápida. Como foi descrito na seção 2.3, a utilização de banco de dados distribuídos possui todas estas características e pode ser uma boa opção para este caso. Entretanto, sua implementação é bastante complexa já que se deve definir, entre outras coisas, como fisicamente os dados devem ser divididos, como as informações devem ser distribuídas,



além de definir regras de segurança e se para a aplicação em questão é interessante ou não utilizar replicação dos dados.

A definição de uma proposta de distribuição de dados deve ser totalmente voltada ao modelo de aplicação, ou seja, não existe uma definição ideal para todos os casos. Deve-se então descobrir qual o nível de distribuição e como ela deve ser feita para a aplicação em questão. Em muitos casos pode-se chegar a conclusão que a distribuição é inviável (por motivos financeiros ou de falta de infra-estrutura) ou desnecessária (custos com recursos extras não compensam as melhorias obtidas).

Como o objetivo deste trabalho não é discutir qual a melhor opção de integração de bancos de dados, não serão indicadas soluções genéricas, apenas serão citadas algumas opções que dependendo das características de cada aplicação, devem ou não ser adotadas.

Nos próximos dois capítulos serão mostrados as principais vantagens e desvantagens da utilização de banco de dados distribuídos.

#### 5.4 VANTAGENS DO BANCO DE DADOS DISTRIBUÍDO

De acordo com (OZSU, 1999), existem várias vantagens na implantação de SGBDD's, das quais podemos destacar:

- a) ***gerenciamento da distribuição com diferentes níveis de transparência*** - um SGBDD pode proporcionar transparência na distribuição onde os detalhes de como os objetos (tabelas, relações, *views*, etc) são distribuídos, são fisicamente armazenados no sistema, dando ao usuário e/ou suas aplicações, transparência total sobre sua localização. Os tipos de transparência necessitam de alguns cuidados especiais à sua implantação para manter a estabilidade do banco;
- b) ***autonomia local*** - através do uso de SBDDs, é possível acessar os dados localizados em *sites* próximos aos usuários que mais os utilizam e, ao mesmo tempo, compartilhar dados armazenados em outros *sites*. Em organizações descentralizadas, o gerenciamento dos dados pode ser delegado aos administradores locais, permitindo maior autonomia e responsabilidade;

- c) **melhor performance** - como os dados estão mais próximos ao usuário que os mantêm, a perda de velocidade no processamento em rede é diminuída. Outro fator é que, como os dados foram divididos e cada porção da tabela ficou menor, o esforço de processador (ou CPU) e E/S (comunicações de entradas e saídas) é bem menor. Pensando em tabelas distribuídas, uma transação pode ser “quebrada” e ser executada em vários servidores, possibilitando sua execução em paralelo;
- d) **melhor integridade e disponibilidade** - se um dado é replicado e, portanto, está disponível em mais de um *site*, em caso de alguma falha em um dos computadores servidores, é possível configurar o SBDD para que o dado esteja sempre disponível;
- e) **economia** - diminui-se o custo com comunicação de dados, uma vez que os dados estão mais próximos dos usuários. Özsü afirma que o ganho é relevante nas situações onde o volume de tráfego é tarifado. Os gastos com aquisição de servidores também podem ser reduzidos. Esta redução se deve ao fato de que computadores de menor porte podem ser utilizados, tendo em vista que o volume de processamento é menor para os dados locais;
- f) **expansibilidade** - ao utilizar um SBDD, os esforços e custos associados ao aumento no número de *sites* usuários em uma rede podem ser menores quando comparados com os custos associados à expansão de um banco de dados centralizado; e
- g) **compartilhamento** - em organizações geograficamente distribuídas e muito distantes entre si, a distribuição de dados é necessária, já que mantê-los centralizados pode envolver um custo muito alto. Os dados distribuídos podem ser compartilhados por vários locais a um custo bem inferior.

## 5.5 DESVANTAGENS DO BANCO DE DADOS DISTRIBUÍDO

De acordo com (OZSU, 1999), existem algumas desvantagens na implantação de SGBDD's, das quais podemos destacar:

- a) **inexperiência** - SBDD's ainda não são muito utilizados. Existe uma carência de especialistas nas unidades responsáveis pelo gerenciamento de tecnologia de

informação em várias organizações. Como consequência, vários problemas podem ocorrer durante a implantação e manutenção de um banco de dados distribuídos;

- b) **complexidade** - os banco de dados distribuídos são mais complexos que os bancos de dados centralizados, principalmente nos aspectos de modelagem, processamento de consultas, concorrência, sistemas operacionais;
- c) **custo** - existe um aumento nos custos associados a treinamento, administração do ambiente operacional (técnicas, ferramentas e pessoal) e desenvolvimento de aplicativos;
- d) **distribuição do controle** - apesar de ser considerado anteriormente como um benefício de um SBDD, autonomia local cria problemas de coordenação e sincronização. O controle exige mais responsabilidade e policiamento dos administradores dos bancos de dados; e
- e) **segurança** - é um dos maiores benefícios dos bancos centralizados. Entretanto, em um banco distribuído, garantir a segurança de acesso para todos os clientes da informação é uma tarefa complexa.

## 5.6 VANTAGENS DA INTEGRAÇÃO ENTRE BDD E GERÊNCIA DE REDES

Além das vantagens de se integrar as bases de dados de uma empresa distribuída, podemos somar outros benefícios se utilizarmos esta tecnologia integrada com gerência de redes.

Se todas as filiais de uma empresa estiverem integradas em um banco de dados distribuídos, independente da forma e do nível de distribuição, pode-se consultar os dados de gerência (como os demais dados da empresa) de forma rápida e fácil. Este recurso pode facilitar em muito o trabalho do administrador, já que ele pode saber como está o funcionamento da rede de toda a empresa de forma quase que imediata e pode consultar dados já arquivados, o que minimiza a redundância de erros.

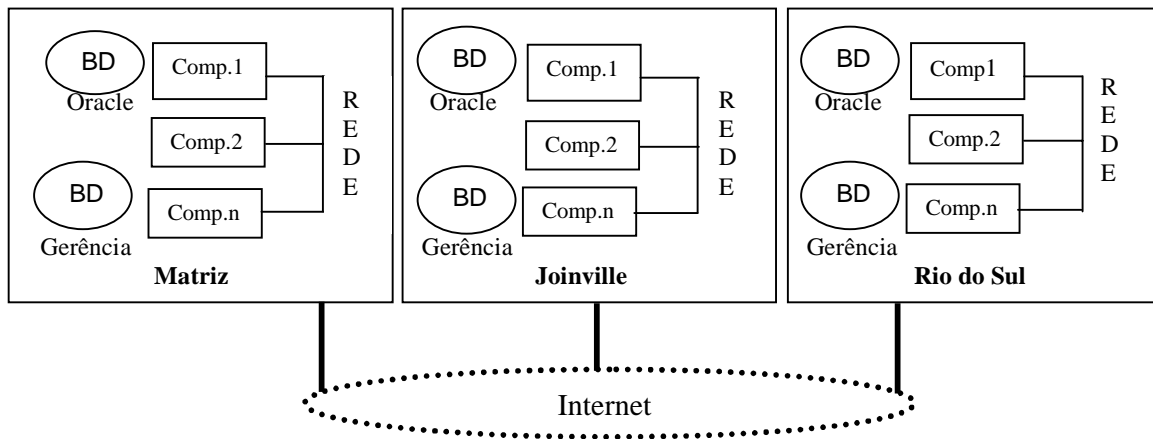
Outro benefício seria que como os dados estão integrados em um banco de dados “conhecido” podem ser desenvolvidas aplicações que utilizam uma interface amigável para ajudar o administrador, mostrando os dados de gerência em gráficos, relatórios, etc. Uma

vantagem da utilização de um banco de dados “conhecido” é que consultas diretas aos dados da filial local ou remota podem ser feitas utilizando para isso linguagem *sql* ANSI padrão, linguagem esta geralmente conhecida pelos DBA’s.

Um outro recurso interessante é que se pode replicar os dados de gerência em um ou vários sítios. Desta maneira estas informações podem ser acessadas mesmo que a conexão com a filial de onde se originam os dados esteja com problemas. Ou mesmo, pode-se utilizar replicação de dados para deixar o processo de consulta mais rápido. Neste caso as cópias devem ser feitas em sites onde utilizem estes dados com grande frequência (no caso das informações de gerência a matriz seria uma boa opção).

A integração de banco de dados distribuídos e gerência de redes agrega vantagens das duas tecnologias, isso torna sua utilização competitiva perante outras soluções usadas pelo mercado.

### 5.7 AMBIENTE DE IMPLEMENTAÇÃO



**Figura 9** - Ambiente de implementação

Neste trabalho será desenvolvido um protótipo de uma empresa formada de uma matriz em Florianópolis, uma filial em Joinville e outra em Rio do Sul. Trata-se de um ambiente de três sítios geograficamente separados, onde cada unidade possui uma rede heterogênea com objetos gerenciáveis. Deve-se salientar que as redes das filiais não são homogêneas entre si, ou seja, existem objetos ou mesmo produtos de marcas diferentes em cada filial.

Todos os sítios usam o mesmo banco de dados e possuem o mesmo modelo de dados. Foi escolhido o Oracle versão 8.0.5.0, pois se trata de um produto renomado na comunidade internacional, tendo suporte a banco de dados distribuído, além disso, tenho boa experiência na utilização deste produto e facilidade para fazer testes em meu ambiente de trabalho.

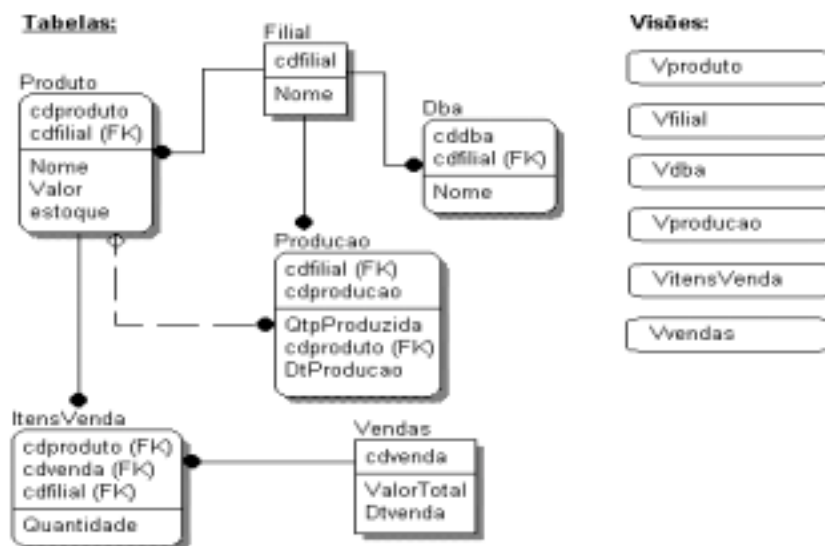
Em cada sítio são usadas regras de segurança próprias definidas pelo DBA local. No protótipo implementado neste estudo este aspecto será ignorado já que não faz parte do foco principal deste trabalho.

Em todos os sítios existe uma linha dedicada com a Telesc de 64kb/s. Esta linha é usada somente para a utilização da internet (www, e-mail, etc). Porém, depois da integração dos bancos de dados ela será utilizada também para fazer a conexão entre a matriz e suas filiais. Com isso pode-se interligar de forma permanente (24 horas por dia, 7 dias por semana) todos os sítios da empresa.

Em cada sítio é utilizado um software de gerência que obtêm os dados dos objetos gerenciáveis da rede e armazena estas informações em uma base de dados proprietária.

Através da Figura 9 deve-se notar que apesar de existir uma possível ligação via internet, no ambiente proposto não existe nenhuma conexão ou troca de dados entre os sítios da empresa.

## 5.8 MODELO DE DADOS DOS SÍTIOS



**Figura 10 - Modelo de dados dos sítios (antes da distribuição dos dados)**

Como se pode notar através da Figura 10, neste protótipo o modelo de dados utilizado na base de dados Oracle de cada sítio da empresa possui apenas as entidades:

- Produto - cadastro dos dados dos produtos;
- Filial - cadastro dos dados da filial;
- Dba - cadastro dos DBA's da filial;
- Vendas - cadastro dos dados das vendas da filial;
- ItensVenda - cadastro dos dados dos itens das vendas da filial; e
- Produção - cadastro dos dados dos produtos produzidos na filial.

Foi definido que para ter maior segurança e facilidade de manutenção, as aplicações não devem fazer acessos diretos às tabelas de dados. Por este motivo, para poder consultar estas entidades, foi criado uma visão para cada tabela. Tem-se então as seguintes visões:

- Vproduto - visão dos dados dos produtos;
- Vfilial - visão dos dados da filial;
- Vdba - visão dos dados dos DBA's da filial;
- Vvendas - visão dos dados das vendas da filial;
- VitensVenda - visão dos dados dos itens das vendas da filial; e
- Vproducao - visão dos dados dos itens das vendas da filial.

Deve-se notar que os dados de gerência não aparecem no modelo de dados, já que estas informações são armazenadas em um banco de dados independente.

Para que se possa ter um banco de dados distribuído deve-se primeiramente ter um banco de dados local integrado. Na próxima seção será descrita uma proposta para fazer esta integração.

## 5.9 INTEGRAÇÃO DOS DADOS DENTRO DA FILIAL

Para que se possa ter um banco de dados distribuído, primeiramente deve-se integrar todos os dados sobre cada sítio dentro do mesmo banco de dados. Como foi descrito nas seções anteriores, as informações de gerência do ambiente de implementação estão armazenadas em um banco de dados independente e desta maneira não podem ser acessadas diretamente.

Para poder obter esta integração, em cada sítio foram criadas as tabelas 'DGR' e 'OBJ\_GR'. A primeira será usada para armazenar todas informações sobre gerência de redes, já a segunda armazenará as informações sobre os objetos gerenciáveis da rede. Entretanto, como os dados não estão armazenados no banco de dados Oracle, devemos criar algum mecanismo de obtenção destes dados.

Como a maioria dos softwares de gerência não possuem um processo de exportação direta para um banco de dados externo, a solução é desenvolver um programa de importação de dados, ou seja, este novo aplicativo deve obter os dados do software de gerência e gravar estas informações no banco de dados Oracle.

Para obter estes dados de gerência existem duas opções. A primeira seria desenvolver um aplicativo para consultar os dados direto dos objetos gerenciáveis da rede (usando SNMP ou CMIP). Este seria um processo complexo já que deve ser implementado toda a parte de comunicação e consulta aos objetos. A segunda opção seria criar uma interface de comunicação para consulta dos dados da ferramenta de gerência. Esta opção é mais simples que a anterior, já que geralmente as ferramentas de gerência possuem alguma forma de acesso direto ou alguma opção de exportação de dados (para arquivo texto, xml, etc).

O gerente da rede (local ou global) deve definir quais as informações de gerência que devem ser armazenadas.

Uma outra utilização interessante deste novo aplicativo é que ele pode criar mecanismos de geração de alarmes e tratamentos de falhas, ou seja, quando ocorre uma falha (e/ou erro) este aplicativo informa esta ocorrência ao gerente (local ou global) indicando os dados necessários e a rede da filial em questão. Isso pode agilizar o processo de resolução do problema.

## 5.10 DISTRIBUIÇÃO DOS DADOS

Como foi citado no capítulo 2.3, a maneira como os dados são distribuídos num SBDD deve ser escolhida dependendo da aplicação em questão.

Como o enfoque principal deste trabalho é a gerência de redes, foi definido que os dados de gerência (que estão guardados nas tabelas 'DGR' e 'OBJ\_GR') das filiais seriam replicados na matriz. Este processo torna o acesso a estes dados mais seguro (pois se a conexão com a filial tiver problemas os dados ainda estarão disponíveis para consulta) e rápido (já que a matriz deve consultar estas informações com grande frequência).

Mesmo a fragmentação horizontal não sendo totalmente implementada no SGBD Oracle, pode-se usar o conceito *database links* para “agrupar” uma informação que está distribuída em vários bancos de dados. Por exemplo, se em cada filial tivermos a tabela “produto” (que contém os dados dos produtos) e se quisermos obter o nome de todos os produtos da empresa teríamos que “agrupar” a informação “nome do produto” das tabelas “produto” das três filiais e utilizarmos o conceito de *database link* para fazer a interconexão entre os bancos de dados.

Para ilustrar este conceito, no protótipo será criada uma visão no banco de dados da matriz chamada *vNomesProdutos* que consultará o nome de todos os produtos da empresa.

Estrutura da visão.

```
Ex.:create view vNomesProdutos as
select nome from vproduto
Union select nome from vProduto@Joinville
Union select nome from vProduto@Riodosul
```



Algumas informações que anteriormente à distribuição estavam armazenadas em todos os sítios, agora podem ser guardadas em apenas uma filial. Por exemplo, se temos a tabela ‘Dba’ em cada sítio, com a distribuição podemos escolher apenas uma filial para guardar estas informações e usar o conceito de *database link* para acessar estes dados. Isso é usado para diminuir custos, diminuir a redundância ou em tabelas com informações que são pouco acessadas. O sítio escolhido para conter os dados deve ser o que mais acesse estas informações. Neste caso, se uma filial remota que possua permissão quiser acessar este dado, ela também deve usar uma visão (que internamente utiliza um ou vários *database link*).

Para ilustrar este conceito, no protótipo será criado uma visão no banco de dados da filial Joinville chamada *vNomesDbas* que consulta os nomes dos DBA’s da empresa que estão cadastrados na tabela “Dba” na matriz.

Estrutura da visão.

Ex.: create view vNomesDbas as

```
select nome from vDba@matriz
```

Note que com a distribuição, o modelo de dados dos sítios foram modificados e agora não são mais homogêneos.

### 5.11 MODELO DE DADOS (DEPOIS DA DISTRIBUIÇÃO DE DADOS)

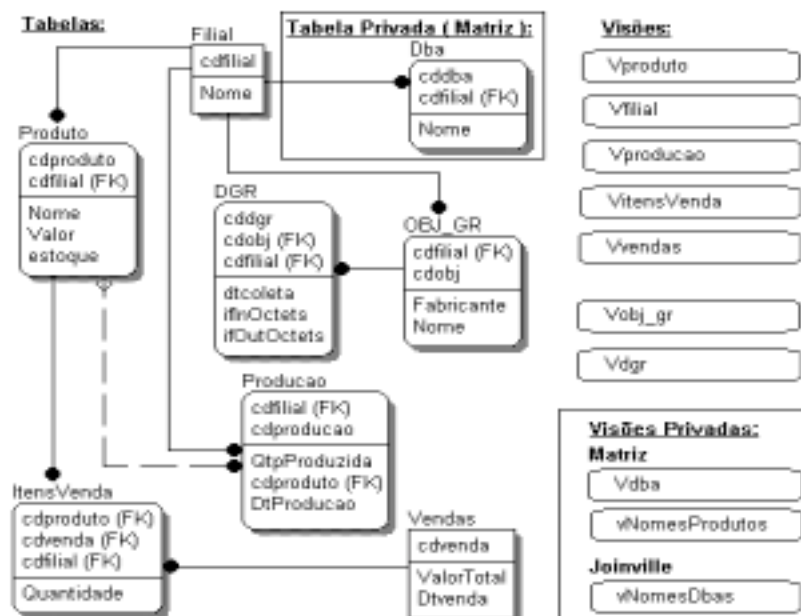


Figura 11 - Modelo de dados dos sítios (depois da distribuição dos dados)

- DGR - cadastro dos dados de gerência do sítio; e
- OBJ\_GR - cadastro dos dados dos objetos gerenciáveis do sítio.

As visões necessárias para consultar os dados de gerência não são mais homogêneas entre os sítios, pois a matriz consulta os dados de toda a empresa, já as filiais consultam apenas as informações referentes a sua unidade.

Então para a matriz teremos as visões:

```
create view vDGR as
select cdfilial,cdobj,datahora,ifInOctets,ifOutOctets from DGR
Union cdfilial,select cdfilial,cdobj,datahora,ifInOctets,ifOutOctets from
vDGR@Joinville
Union select cdfilial,cdobj,datahora,ifInOctets,ifOutOctets from vDGR@Riodosul
```

```
create view vOBJ_GR as
select cdfilial,cdobj,nome,fabricante,modelo from OBJ_GR
Union select cdfilial,cdobj,nome,fabricante,modelo from vOBJ_GR@Joinville
Union select cdfilial,cdobj,nome,fabricante,modelo from vOBJ_GR@Riodosul
```

Para as filiais teremos as visões:

```
create view vDGR as
select cdfilial,cdobj,datahora,ifInOctets,ifOutOctets from DGR
```

```
create view vOBJ_GR as
select cdfilial,cdobj,nome,fabricante,modelo from OBJ_GR
```

Como é possível observar, depois da distribuição o modelo de dados dos sítios não é mais homogêneo. Além da adição das tabelas que armazenam os dados de gerência de redes (tabelas 'DGR' e 'OBJ\_GR'), o número de entidades e como estas estão distribuídas nas unidades difere de um sítio para outro.

Depois da distribuição, a matriz é o sítio que possui o número maior de entidades. Isso já era esperado, pois esta unidade controla as demais filiais em todos os aspectos (financeiro, comercial, gerência de redes, etc). Já a filial Rio do Sul é o sítio que tem o número menor de entidades, que difere da Filial Joinville, pois não possui a visão vNomesDbas. Isso ocorre porque esta informação não é utilizada pelas aplicações existentes nesta filial, ou seja, o modelo de dados depende diretamente de quais as informações e como elas são usadas pelas aplicações em cada sítio.

Deve ficar claro que o modelo de dados também pode ser modificado dependendo da necessidade do DBA de cada sítio, ou seja, o administrador continua podendo criar entidades para armazenar e/ou consultar dados interessantes somente a ele e que não existam no modelo de dados atual. Com isso a distribuição não torna o trabalho do DBA menos flexível, apenas em alguns casos pode ser usada para controlar melhor suas funções. Isso vale também para as informações de gerência.

Com a distribuição, criou-se a necessidade de comunicar os sítios entre si. Para que isso funcione de forma adequada foram utilizados novos recursos que serão descritos nas duas próximas seções.

## 5.12 CONEXÃO ENTRE AS FILIAIS

Para que se possa consultar dados distribuídos deve-se utilizar algum mecanismo que faça a conexão entre o sistema local e o remoto. Em banco de dados distribuídos esta conexão deve ser transparente ao usuário e obter as respostas das demandas da maneira mais rápida possível.

No Oracle pode-se utilizar o conceito de *database link* para fazer esta conexão, porém não se trata de uma solução ideal a nível de banco de dados distribuídos, pois a utilização desta ferramenta deixa a critério da aplicação escolher qual *database link* irá acessar.

Para o protótipo que foi implementado neste trabalho foram criados os seguintes *database link*:

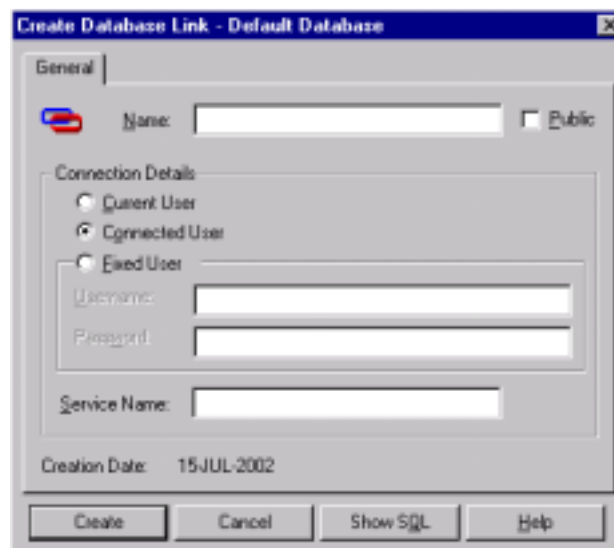
Na matriz foram criados os *database link* Riodosul e Joinville. O primeiro é utilizado para acessar o banco de dados da filial Rio do Sul, já o segundo é usado para acessar o banco

de dados de Joinville. Como a matriz tem o papel de controlar as demais filiais ela deve ter acesso ao banco de dados de todos os sítios.

Na filial Joinville foi criado apenas um *database link* utilizado para consultar dados dos DBA's da empresa que estão armazenados na matriz. Não existe uma conexão entre a filial Joinville e a Rio do sul, neste caso isso é interessante, pois as aplicações utilizadas em Joinville não devem manejar informações de Rio do Sul.

Na filial de Rio do Sul não foi criado nenhum *database link*, ou seja, esta unidade não consulta informações de sítios externos, porém pode fornecer dados a eles. Esta conexão de via única é muito interessante por motivos de segurança, já que a filial não pode acessar ou modificar informações que não sejam de seu interesse, além de tornar as respostas de consultas externas mais rápidas, pois assim ela utiliza a linha de comunicação apenas para respostas, o que gera menos tráfego.

O Processo de criação de um *database link* no Oracle é bem simples. Em uma interface agradável o DBA escolhe um nome (que será usado para identificá-lo), digita os dados para a conexão com a base de dados do sítio remoto (nome do usuário, senha do usuário e string de conexão) e especifica se este *database link* será público ou privado. A figura 12 mostra como é simples e intuitiva a interface de criação de *database link* no Oracle.



**Figura 12** - Interface de criação de *database link*

Porém, antes de decidir se é necessário criar um *database link*, o DBA deve levar em consideração os itens abaixo:

- a) se já existe outro *database link* na base de dados local que acessa o mesmo banco de dados;
- b) se alguma aplicação utilizada localmente irá acessar algum dado que esteja armazenado fisicamente neste sítio;
- c) aspecto de segurança, ou seja, se é possível acessar as informações deste sítio sem gerar problemas de segurança para a empresa; e
- d) se não existe uma réplica das informações na base de dados local ou em outro sítio alcançável por algum *database link* já pertencente a esta base de dados.

Como é possível notar, a replicação de dados pode ser uma boa opção em banco de dados distribuídos. Na próxima seção será descrito o funcionamento desta tecnologia no Oracle.

### 5.13 REPLICAÇÃO DOS DADOS

Como foi citado na seção 2.3, as razões básicas para o uso de réplicas são a disponibilidade dos dados e o melhor desempenho das aplicações. Este recurso pode ser muito interessante, pois torna o acesso aos dados mais rápido e seguro.

Como o enfoque deste trabalho é gerência de redes, foi definido que os dados de gerência (que estão guardados na tabela 'DGR') das filiais seriam replicados na matriz. Nesta seção será descrito como implementar replicação de dados no Oracle.

Primeiramente, deve-se exportar os dados da tabela 'DGR' de cada filial e os inserir na base de dados da matriz. Desta maneira o banco de dados da matriz estará atualizado, ou seja, possuirá todos os dados de gerência da empresa.

O segundo passo é em cada filial criar um *SnapShot* de log das movimentações que ocorreram nesta tabela, este log será usado na matriz para a identificação de quais dados foram modificados, apagados etc...

Ex.: create snapshot log on DGR;

O terceiro passo é em cada filial criar um sinônimo para a tabela 'DGR', desta maneira é possível acessar os dados do log da tabela 'DGR' de uma maneira indireta. Além disso, deve-se tornar este elemento público, para que qualquer usuário do banco possa acessar estas informações. Isso será útil nos próximos passos.

```
Ex: create public synonym MLOG$_DGR
    for LOG$_DGR;
    grant all privileges on MLOG$_DGR to public;
```

O quarto passo é criar na matriz um *SnapShot* de atualização dos dados (de replicação) da tabela 'DGR' para cada filial replicada, ou seja, cada *SnapShot* tem como objetivo tirar uma 'foto' dos dados da tabela 'DGR' da filial e através da comparação dos dados locais com o *log* de transações atualizar as informações de gerência (Utilizando para isso *Trigger's*).

Para tornar este processo automático, pode-se fazer com que ele se auto-execute em períodos de tempo configuráveis. No exemplo descrito abaixo, o *SnapShot* que atualiza os dados de gerência da filial de Rio do Sul ('RS\_DGR') está programado para se auto-executar uma vez a cada dia.

```
Ex.: create snapshot RS_DGR
    refresh force start with sysdate + 4/24 next sysdate + 1 as
    select * from DGR@ Riodosul;
```

O quinto passo é um complemento do passo anterior, ou seja, para que se possa atualizar os dados de gerência, deve-se adicionar em cada *SnapShot* criado na matriz um *Trigger* que faça essa atualização. Desta maneira, a cada vez que o *SnapShot* for executado é verificado através do log se ocorreram mudanças na tabela e estas alterações são replicadas na tabela da matriz (Usando para isso este *Trigger*). Para o *SnapShot* da filial de Rio do Sul teríamos o seguinte *Trigger*.

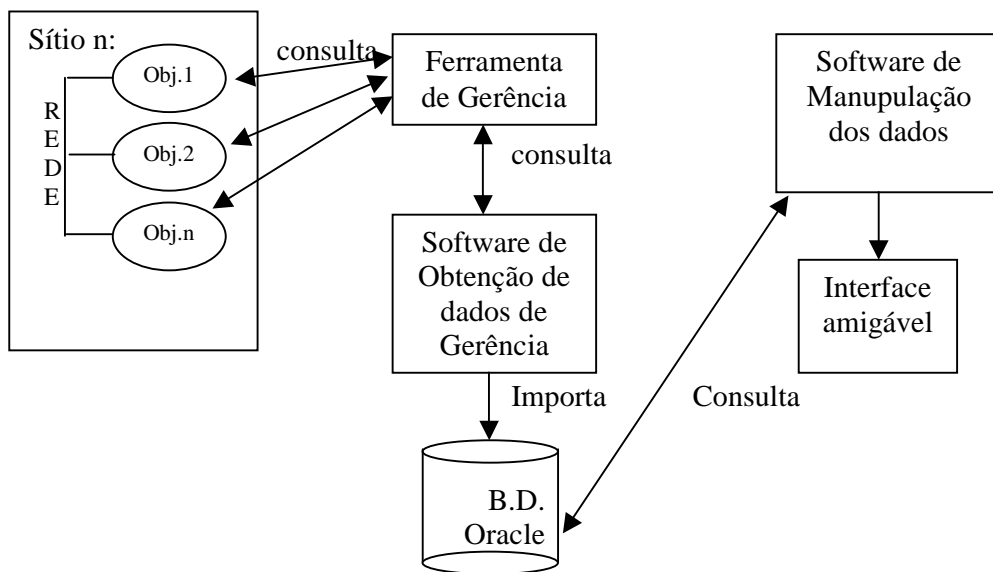
```
Ex.: CREATE OR REPLACE TRIGGER T$_RS_DGR
      AFTER INSERT OR UPDATE OR DELETE
      ON SNAP$_RS_DGR
      FOR EACH ROW
      BEGIN
        IF INSERTING THEN
          insert into DGR(CD_CAMPO)
            values (:NEW.CD_CAMPO);
        ELSIF UPDATING THEN
          update DGR
            set CD_CAMPO = :NEW.CD_CAMPO;
        ELSIF DELETING THEN
          delete from DGR
            where CD_CAMPO = :OLD.CD_CAMPO;
        END IF;
      END;
```

O último passo seria inicializar todo esse processo de replicação. Depois desta chamada inicial o processo pode ser usado pelo servidor como serviço e se auto-executar indefinidamente.

```
Ex.: exec dbms_refresh.refresh(' RS_DGR ');
```

Como se pode observar, o processo de replicação no Oracle depende da utilização de uma série de ferramentas, porém sua implementação é relativamente fácil e pode ser uma boa alternativa para muitas aplicações.

## 5.14 FUNCIONAMENTO DO PROCESSO DE GERÊNCIA



**Figura 13** - Funcionamento do processo de gerência

Nesta seção será descrito o funcionamento do processo de gerência de redes que é o enfoque principal deste trabalho. Este processo utiliza basicamente três softwares:

- a) **ferramenta de gerência** – é “invisível” a nível de usuário. Trata-se de um aplicativo que geralmente roda no servidor e que pode fazer consultas aos objetos gerenciáveis da rede com o objetivo de obter informação sobre estes objetos as quais serão utilizadas pelos administradores;
- b) **software de obtenção dos dados de gerência** - também é “invisível” a nível de usuário. Trata-se de um aplicativo que roda como serviço no servidor da rede e que em períodos de tempos configuráveis consulta as informações que foram obtidas pela ferramenta de gerência e as grava no banco de dados Oracle; e
- c) **software de manipulação dos dados** - com os dados já devidamente armazenados, pode-se utilizar este software com interface amigável para consultar estas informações da maneira mais rápida, segura e de forma transparente para o usuário. Assim é possível manipular estes dados da forma desejada. Este software pode ser criado



somente com esta finalidade, ou pode-se adicionar esta função de gerência de rede a alguma outra aplicação já utilizada pela empresa.

Deve ficar claro que internamente os softwares devem utilizar técnicas de banco de dados distribuídos, entre elas podemos citar o uso de fragmentação e replicação de dados (técnicas já descritas em seções anteriores deste trabalho) a fim de obter melhor performance e segurança. Porém essas técnicas devem ser “invisíveis” aos olhos do usuário.

O funcionamento do processo de gerência é bastante simples. No primeiro passo a ferramenta de gerência consulta os dados dos objetos gerenciáveis da rede que estão cadastrados na tabela ‘OBJ\_GR’, então o software de obtenção de dados de gerência consulta as informações vindas desta ferramenta e as importa para o banco de dados Oracle. Em seguida os dados são gravados na tabela ‘DGR’, e com as informações já armazenadas, o software de manipulação dos dados pode disponibilizá-las de maneira mais interessante para o administrador, podendo utilizar para isso gráficos, relatórios etc, além de poder usar outros recursos de banco de dados tais como filtros, agrupamentos ou ordenação.

Então, para tomar alguma decisão de gerência o administrador entra no Software de Manipulação de Dados, consulta os dados que lhe interessam e obtêm as informações necessárias de forma rápida, segura e numa interface agradável.

Nesta seção foi descrito como funciona o processo de gerência. É possível notar que se trata de um processo simples e que pode obter bons resultados com baixo custo.

## 5.15 FUNCIONAMENTO E VALIDAÇÃO DO PROTÓTIPO

O projeto inicial deste trabalho definia a implementação e teste de todos os itens necessários para o processo de integração entre as tecnologias de banco de dados distribuídos e gerência de redes, com o objetivo de validar esta integração e se possível comparar esta nova ferramenta com as demais existentes no mercado.

Porém, nem todos os itens puderam ser desenvolvidos até o término deste trabalho. Então foi definido que seria desenvolvido um protótipo utilizando dados reais, num ambiente real, porém sem desenvolver as duas aplicações utilizadas no processo. Por questão de prioridade foi dado maior enfoque a validação das técnicas de banco de dados distribuídos

utilizadas para armazenar as informações de gerência e facilitar a vida dos administradores. Com isso, detalhes da implementação das aplicações não foram tratadas neste trabalho, já que seu conhecimento é vastamente divulgado no mundo acadêmico.

No protótipo implementado, foi integrado os dados de uma empresa formada de três sítios geograficamente dispersos (um em Florianópolis, um em Rio do Sul e outro em Joinville), cada uma com informações de gerência (além de informações gerais da unidade) de sua rede.

A obtenção de dados em cada sítio foi feita a cada cinco minutos pelo software de gerência “*AdventNet Snmp Utilities Versão 3 release 3.3*”, o qual consulta dados dos objetos gerenciáveis de sua rede (*Switchs, hubs, etc*). Então estas informações eram obtidas da ferramenta de gerência e exportadas para a tabela ‘DGR’ do banco de dados Oracle local. O funcionamento completo deste processo será descrito na próxima seção.

Foi implementado tanto o processo de fragmentação descrito na seção 5.10 como o processo de replicação de dados descrito na seção 5.13. Para este último foi configurado um período de 10 minutos de auto-execução.

Não foi implementado o software de manipulação de dados. Entretanto, para validar este processo foi feita uma simulação, que através do uso de sql padrão ANSI faz consultas ao banco de dados para obter as informações de gerência. Deve ficar claro que neste caso existem duas opções de consulta: Direta e Indireta. É utilizada a primeira opção se o objetivo é obter as informações naquele “exato” momento (dados de resposta com defasagem de no máximo cinco minutos), neste caso é interessante consultar os dados direto dos sítios, utilizando para isso fragmentação. Entretanto, se tiver na matriz e quiser obter respostas com maior agilidade, é interessante utilizar a segunda opção, onde as consultas são feitas direto ao banco de dados local (dados de resposta com defasagem maior ou igual há dez minutos). Como ficou evidenciado nesta seção, a escolha de qual opção deve ser utilizada depende da urgência e para qual uso será feita estas informações, ficando a critério da aplicação decidir qual a melhor alternativa.

Contudo, se ocorrerem problemas na conexão, não será possível consultar os dados diretamente da origem. Neste caso, a consulta indireta passa a ser a única escolha. Neste trabalho também foram simulados problemas desta natureza. Os resultados obtidos com os testes serão descritos na seção 5.18.

## 5.16 VALIDAÇÃO DO PROCESSO DE OBTENÇÃO DOS DADOS DE GERÊNCIA

Nesta seção será descrito como foi validado o processo de obtenção de dados de gerência.

Como o objetivo deste trabalho é apenas validar o processo de integração, foram escolhidas apenas duas informações de gerência para ilustrar esta validação. Porém deve ficar claro que em um ambiente real de uma empresa estas informações seriam insuficientes para um administrador gerenciar adequadamente sua rede.

Foram utilizados apenas as variáveis SNMP *ifInOctets* e *ifOutOctets*. A primeira informa a quantidade de bytes que entra na interface de cada objeto, já a segunda informa a quantidade de bytes que sai desta interface. Ambas as variáveis pertencem a MIB-II e são identificadas por:

*IfInOctets*

.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifInOctets  
.1.3.6.1.2.1.2.2.1.10

*IfOutOctets*

.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutOctets  
.1.3.6.1.2.1.2.2.1.16

Estas variáveis foram escolhidas pois são muito utilizadas em administração de gerência de redes. Além disso, elas serão utilizadas no protótipo implementado neste trabalho para calcular o tráfego de dados em cada objeto num determinado horário. O funcionamento deste cálculo é simples. Inicialmente é obtido o valor atual das variáveis. Em seguida, diminui-se este valor do valor da última consulta e por fim divide o valor obtido pelo tempo decorrido entre as consultas (em segundos). Isso terá como resultado o tráfego de entrada e saída de cada objeto gerenciável. As informações de *ifInOctets* e *ifOutOctets* serão gravados na tabela 'GR'.

Para o protótipo implementado foram escolhidos três objetos gerenciáveis em cada sítio, dos quais serão obtidas as informações de gerência.

Na matriz foram selecionados o *Switch* principal que será identificado pela chave 'M1' na tabela 'OBJ\_GR' (cadastro dos objetos gerenciáveis), e dois *hubs* que serão identificados pelas chaves 'M2' e 'M3'. O primeiro é usado para conectar os micros do centro de processamento de dados da unidade, já o segundo é utilizado no departamento de pessoal.

Na filial de Joinville foram selecionados o *Switch* principal que será identificado pela chave 'J1', e dois *hubs* que serão identificados pelas chaves 'J2' e 'J3'. O primeiro é usado no centro de processamento de dados da filial, já o segundo é utilizado no departamento de contabilidade.

Na filial de Rio do sul foram selecionados o *Switch* principal que será identificado pela chave 'R1', e dois *hubs* que serão identificados pelas chaves 'R2' e 'R3'. O primeiro é usado no centro de processamento de dados da filial, já o segundo é utilizado no departamento de produção.

Como não foi possível desenvolver um software de obtenção dos dados de gerência, este processo foi todo feito de forma manual, ou seja, em períodos de tempo de cinco minutos os objetos cadastrados eram consultados e os resultados eram inseridos na base de dados Oracle. Esse processo foi repetido durante cinco dias em três períodos diários (Nove horas da manhã, uma hora e meia da tarde e seis horas da tarde) durante trinta minutos em cada período, o que gerou dezoito coletas de dados por dia.

Foi utilizado o software de gerência "*AdventNet Snmp Utilities Versão 3 release 3.3*" para fazer estas consultas. Foi escolhido este aplicativo pois possui bom manual, fórum para perguntas e é baseado em java, o que o torna compatível com a maioria dos sistemas operacionais. Entretanto, foi uma escolha pessoal, existem outras empresas com produtos similares que podem ser utilizados para o mesmo propósito.

Utilizando os dados obtidos da ferramenta de gerência, foi possível observar o comportamento do tráfego de dados nos objetos consultados. Estas informações serão melhor evidenciadas na próxima seção que ilustra os testes realizados.

## 5.17 TESTES REALIZADOS

Nesta seção será descrito como foram realizados os testes para validação da integração das tecnologias de banco de dados distribuídos e gerência de redes.

Toda a parte de criação e modelagem do banco de dados foi feita utilizando ferramentas visuais que fazem parte do pacote de instalação. Tais ferramentas tornaram o desenvolvimento destes processos mais rápido e fácil, mostrando que o produto Oracle possui entre outras qualidades a facilidade de uso.

Para validar os processos de fragmentação e replicação, foram feitos os seguintes testes:

- a) **fragmentação** - foram criados os *database link* descritos na seção 5.12. Através do uso de sql padrão ANSI foram feitas consultas sobre as visões que internamente utilizam o conceito de *link*; e
- b) **replicação** - foi criado o processo de replicação de dados para as informações de gerência e realizadas consultas utilizando sql padrão ANSI nos dados gerados pela replicação a fim de confirmar sua integridade.

Ainda foram feitas simulações de problemas de conexão, tanto para o processo de fragmentação quanto para a replicação. Em todos os casos, os problemas foram forçados, ou seja, não ocorreram espontaneamente.

Para validar o processo de obtenção de dados de gerência foi implementado o protótipo descrito na seção anterior. O resultado da média de tráfego em cada sítio nos três períodos escolhidos dos cinco dias onde foram colhidos os dados de gerência foi usado como fonte de dados nas Figuras 14, 15 e 16.

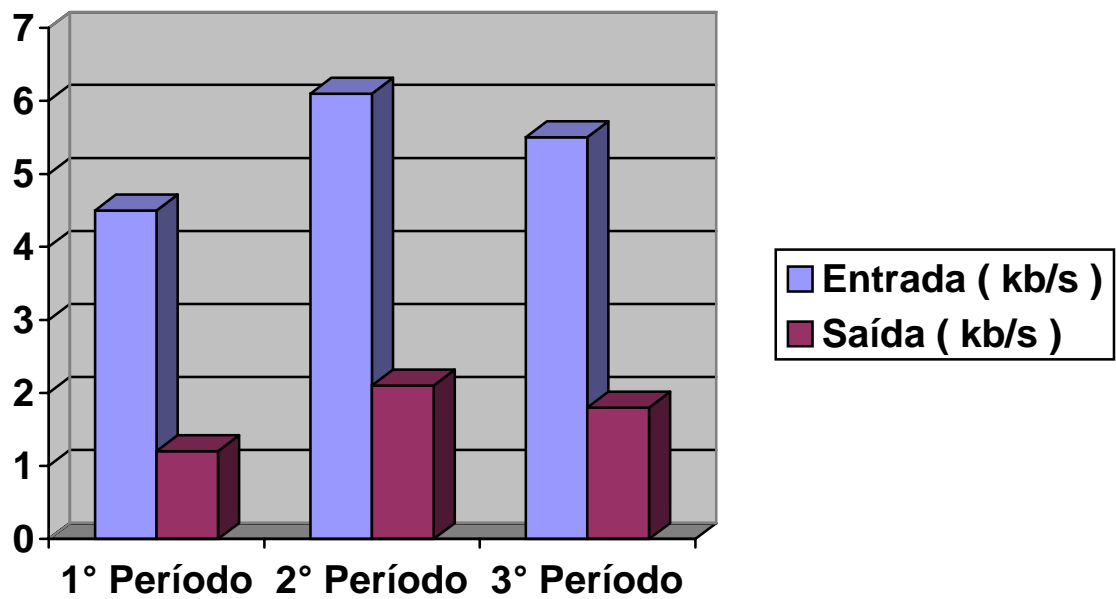


Figura 14 - Tráfego de dados na matriz

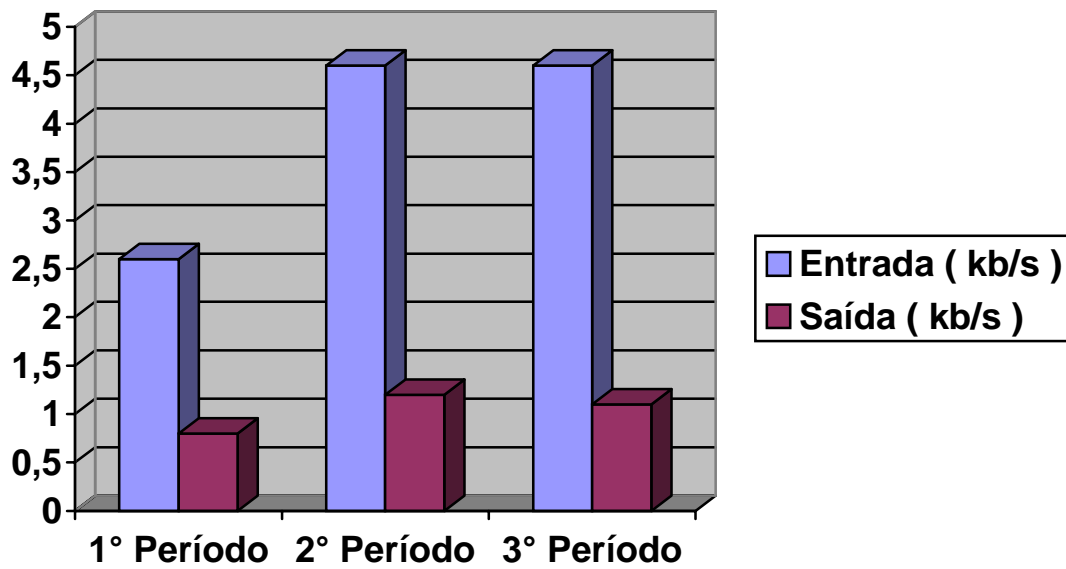
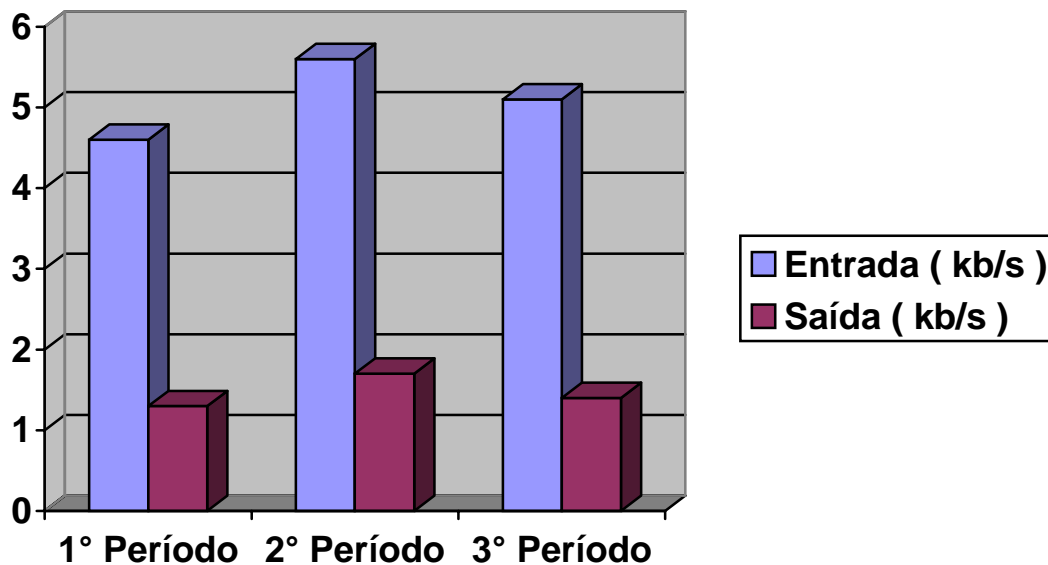


Figura 15 - Tráfego de dados na filial de Rio do Sul



**Figura 16** - Tráfego de dados na filial de Joinville

Ao observar os gráficos gerados, pode-se afirmar que o tráfego de dados nos objetos cadastrados em períodos iguais entre os sítios é bastante similar. Ficou claro que a matriz é o sítio que mais gera tráfego, tanto para *download* quanto para *upload*, chegando a utilizar praticamente toda a banda de internet disponível nos horários de pico. Já a filial de Rio do Sul foi o sítio que obteve em média os menores valores de entrada e saída de bytes. Os dados utilizados para fazer estes gráficos foram obtidos através de consultas diretas, ou seja, neste caso não foram utilizados dados replicados.

Na próxima seção serão descritos os resultados obtidos com os teste realizados neste trabalho.

## 5.18 RESULTADOS OBTIDOS

Os resultados da integração de banco de dados distribuídos e gerência de redes foram satisfatórios, pois os dados obtidos são concisos, organizados e podem ser consultados de forma rápida, fácil e segura.

Nos testes realizados a fragmentação funcionou muito bem, o que mostrou que o conceito de “*link*” utilizado pelo Oracle é válido. O tempo de consulta também foi considerado satisfatório, apesar de que este depende muito de fatores externos, tais como a qualidade da conexão entre os sítios e a velocidade de processamento dos servidores.

Outra tecnologia que teve bons resultados foi a replicação de dados. Os testes feitos provaram que se trata de uma ferramenta robusta e que pode ser uma ótima opção para várias aplicações.

Apesar de não implementar a interface para mostrar os dados de gerência, pode-se dizer que para o administrador da rede a utilização da integração demonstrada pelos testes pode tornar o seu trabalho mais fácil, rápido e eficaz. Um exemplo claro disso são os gráficos de tráfego de dados demonstrados na seção anterior.

Com o protótipo, pode-se observar que a implantação do processo de integração é simples e relativamente fácil. O item de maior dificuldade é a escolha de como distribuir os dados e qual a melhor forma de acessá-los. Outro fator interessante é que se trata de um processo financeiramente viável para a maioria das empresas de médio e grande porte.

Outra situação testada com o protótipo foi o problema de conexão entre os sítios que ocorrem devido a uma série de motivos e que torna a comunicação entre os bancos de dados impossíveis. Neste fator os testes feitos mostraram que se utilizarmos replicação, a integração é uma boa alternativa, pois a ruptura da conexão com um sítio não indisponibiliza as informações dos demais. Além disso, os dados arquivados do sítio com problemas continuam disponíveis para a aplicação. Deve-se citar que algumas tecnologias utilizadas comercialmente para gerência de redes têm dificuldades de lidar com este problema.

Entretanto, as simulações identificaram alguns pontos fracos na integração. Ficou claro neste trabalho que a fragmentação usada pelo Oracle não é ideal, pois a utilização do conceito de *link* deixa a critério da aplicação a escolha de onde se origina os dados e isso



pode ser problemático em algumas aplicações. Além disso, alguns aspectos importantes não foram testados neste estudo, tais como problemas de segurança referente à privacidade, integração de banco de dados heterogêneos, utilização de outros produtos com suporte a banco de dados distribuídos e testes comparativos com técnicas já usadas no mercado.

Como é possível observar, apesar de ter alguns pontos fracos a integração de banco de dados distribuídos e gerência de redes pode ser uma boa opção para empresas de médio e grande porte. Isso é válido para corporações que já possuam um banco de dados com suporte a esta tecnologia ou para novos projetos que desejam integrar as informações de todas as filiais de uma empresa. Este tipo de integração mostrou-se uma grande concorrente para as tecnologias utilizadas pelo mercado atualmente.

## 6 CONCLUSÃO

Para ter um bom domínio sobre as tecnologias utilizadas neste trabalho foi desenvolvido um estudo específico sobre banco de dados distribuídos e seu suporte ao ORACLE.

Ficou claro que a implantação de um banco de dados distribuído é uma tarefa complexa, pois se deve estudar as transações intensivamente para decidir a forma e os critérios de distribuição. Porém, a maior dificuldade se concentra na administração das falhas e recuperação das informações, tarefa que deve ser executada pelo administrador de dados local. Estes problemas, existentes também nos bancos centralizados, são mais complexos e devem ser monitorados constantemente.

Portanto, os administradores devem ser treinados para conhecerem os recursos que o banco de dados (no caso específico, o Oracle) oferece para a administração dessa distribuição.

O SBDD Oracle, por não implantar a distribuição de dados por fragmentação, obriga a aplicação a gerenciar parte da localização das bases, o que resultará em manutenção do atual sistema. Porém, esta manutenção aumentará sensivelmente a velocidade de acesso do sistema e propiciará melhor aproveitamento da capacidade de processamento dos servidores.

Com a distribuição de dados, é possível minimizar os gastos com as redes WAN, o processamento e a capacidade de armazenamento dos servidores, dividindo o esforço de cpu e o tráfego de informações na rede.

O SGBD Oracle já evoluiu muito e, na versão 9i, implementou um esquema de particionamento de tabelas em várias *tablespaces*, o que melhorará a performance no acesso às grandes tabelas. Este particionamento só é possível em um banco de dados centralizado.

Acredita-se que este recurso deva ser estendido para os SBDDs, assim o Oracle teria uma implementação de distribuição por fragmentação.

Através deste trabalho foi possível observar que a utilização das tecnologias de gerência de redes e banco de dados distribuídos podem ser usadas de forma conjunta. Para validar tal processo foram obtidas informações de gerência utilizando o protótipo implementado neste trabalho. Através dos testes realizados ficou claro que a utilização de fragmentação horizontal e replicação como uma ferramenta pelo administrador de redes pode ser uma boa opção, pois os dados resultantes foram concisos e o tempo de resposta foi considerado satisfatório (menos de 10 segundos). Além disso, este tipo de conciliação é viável tanto financeiramente como em termos técnicos. Ela pode ser aplicada em grande parte das empresas de médio e grande porte que pretendem integrar os bancos de dados de suas filiais e que desejam gerenciar suas redes.

Em relação à gerência de redes, ficou claro que a utilização desta tecnologia de integração é uma boa opção, pois através dela pode-se obter as informações de gerência de forma rápida, fácil e segura. Além disso, estes dados podem ser mostrados em interfaces amigáveis através do uso de relatórios, gráficos, etc. O capítulo 5.17 mostra graficamente um exemplo prático, realizado neste trabalho, da utilização deste processo de integração. Ou seja, trata-se de uma boa concorrente para as outras ferramentas disponíveis no mercado.

Tanto a implantação como a manutenção desta solução é relativamente rápida e simples, tanto para empresas que já possuem seu banco de dados e sua rede de dados implementados, como para novos projetos. Deve-se salientar que estes processos são financeiramente viáveis para a maioria das empresas de médio e grande porte.

Como se pode observar através dos testes realizados neste trabalho, esta integração agrega vantagens das duas tecnologias, ao mesmo tempo em que facilita a gerência de redes, simplifica e unifica a organização das informações de uma empresa distribuída.

## 6.1 PRINCIPAIS CONTRIBUIÇÕES

Resumidamente, este trabalho apresentou subsídios que validam as seguintes conclusões.

a) a implantação de um banco de dados distribuído é uma tarefa complexa, pois se deve estudar as transações intensivamente para decidir a forma e os critérios de distribuição. Porém, sua utilização torna as informações de uma empresa distribuída mais integrada;

b) o banco de dados ORACLE não implementa de forma completa a fragmentação horizontal, porém tal solução pode ser utilizada como ferramenta pelo administrador de redes de uma empresa distribuída, obtendo bons resultados;

c) a replicação pode ser utilizada como ferramenta pelo administrador de redes de uma empresa distribuída, obtendo bons resultados. Porém, se a quantidade de informação for muito elevada tal processo torna-se lento;

d) a utilização das tecnologias de gerência de redes e banco de dados distribuídos podem ser usadas de forma conjunta, sendo uma boa concorrente para as outras ferramentas disponíveis no mercado.

e) em relação à gerência de redes, a utilização desta tecnologia de integração é uma boa opção, pois através dela pode-se obter as informações de gerência de forma rápida, fácil e segura;

f) a implantação e a manutenção desta solução de integração é relativamente rápida e simples;

g) custo relativamente baixo da implementação desta tecnologia; e

h) a integração agrega vantagens das duas tecnologias.

## 6.2 TRABALHOS FUTUROS

Este trabalho apenas validou o processo de integração. Seria muito interessante desenvolver o processo como um todo, implementando os aplicativos de obtenção dos dados de gerência e de manipulação de dados de gerência.

Com isso, haveria além da parte funcional uma interface mais amigável, o que facilitaria a demonstração de resultados e a tomada de decisões de gerência.

Neste trabalho não foi validada a parte de segurança referente à privacidade, ou seja, não foram feitos ataques ou simulação de tentativas de obtenção de dados de pessoas não autorizadas. Seria muito interessante simular tais ações e observar como o Oracle trata este problema.

O protótipo implementado neste trabalho teve como base o banco de dados Oracle. Um trabalho futuro poderia utilizar produtos similares de outros fabricantes que possuam suporte a banco de dados distribuídos, verificando performance, aspectos de segurança e controle, além dos pontos fortes e fracos de cada produto.

Em virtude da indisponibilidade do produto mais atual da empresa Oracle, foi necessário a utilização da versão 8.0.5.0. Um trabalho futuro poderia usar esta nova versão 9.i para testar se ela possui alguma vantagem que pode ser aproveitada para obter melhores resultados na integração.

Seria muito interessante fazer um estudo comparativo entre a tecnologia proposta neste trabalho e outras utilizadas no mercado (tais como agentes móveis), verificando desempenho, funcionalidade e confiabilidade, além das vantagens e desvantagens de cada ferramenta.

Ou seja, esta área de estudo possui um grande potencial para trabalhos futuros.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação – referências - elaboração. Rio de Janeiro, 2000.

\_\_\_\_\_. **NBR 6024**: numeração progressiva das seções de um documento. Rio de Janeiro, 1989.

\_\_\_\_\_. **NBR 6027**: sumário. Rio de Janeiro, 1989.

\_\_\_\_\_. **NBR 6028**: resumos. Rio de Janeiro, 1990.

\_\_\_\_\_. **NBR 12256**: apresentação de originais. Rio de Janeiro, 1992.

\_\_\_\_\_. **NBR 14724**: informação e documentação – trabalhos acadêmicos – apresentação. Rio de Janeiro, 2001.

ANEROUSIS N. *A distributed computing environment for building scalable management services*. *Proc. IEEE IM*, 1999, pp. 547 - 562, May 1999.

BAIÃO, Fernanda Araújo. **Distribuição e Processamento de Consultas em banco de Dados Orientados a Objeto**. [S.l.: s.n], 1995.

BAPTISTA, Deoclésio J. V. D. **Da Teoria à Prática – O Processo de Descentralização da Prodabel**. Belo Horizonte: EGMG/FJP, 1998.

BELL, David; GRIMSOM, Jane. *Distributed database systems*. Workingham, England: Addison-Wesley, 1992. 410 p.

BURETTA, M. *Data replication: tools and techniques for managing distributed information*. [S.l.]: Wiley Computer Publishing, 1997.

CARVALHO, T. C. M. B. et al. **Gerenciamento de redes: uma abordagem de sistemas abertos**. São Paulo: Sociedade Brasileira para Interconexão de Sistemas Abertos; Makron Books, 1993.

CASANOVA, Marco Antônio. **Princípios de sistemas de gerência de banco de dados distribuídos**. Rio de Janeiro: Campus, 1985. 355 p.

CASTRO, C. E. P. S. **Integração de Legacy Systems a Sistemas de Bancos de Dados Heterogêneos**. 1998. Dissertação (Mestrado) - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.

CERI, Stefano; PELAGATTI, Giusepper. *Distributed databases principles & systems*. New York: McGraw-Hill, 1992. 410 p.

CERÍCOLA, Osvaldo V. **Banco de Dados Relacional e Distribuído**. São Paulo: Makron Books, 1995.

ELMASRI, Ramez; NAVATHE, Shamkant B. *Fundamentals of Database Systems*. Redwood City: Benjamin/Cummings, 1999.

EVREDILEK C.; DOGAC A.; NURAL S.; OZCAN F. *Multidatabase query optimization. Journal of Distributed and Parallel*. Databases, vol. 5, n. 1, 1997.

FEIT, Sidnie. **SV.W: A guide to network management**. McGraw-Hill series on computer communications. USA: McGraw-Hill Trade, 1995.

FERREIRA, J. E.; FINGER, M. **Controle de concorrência e distribuição de dados: a teoria clássica, suas limitações e extensões modernas**. 2000. Tese (Doutorado) - Programa de Ciência da Computação, Universidade de São Paulo, São Paulo.

GUENGERICH, Steven; DOWNSING R. **Sistemas de Informação**. São Paulo: Makron Books, 1993.

HARITSA J. R.; M. O. Ball ; N. Roussopoulos ; A. Datta ; J. S. Baras. **MANDATE: MANaging Networks Using DAtabase Technology**. *IEEE Journal Selected Areas in Comm.*, vol. 11, pp. 1360 - 1372, December 1993.

HOPPER, S. TERESA, R. *Distributed relational database architecture connectivity guide*. San José USA: IBM Corporation, 1995. 477 p.

INTERNATIONAL BUSINESS MACHINES (USA). *Distribuído relational database architecture*. San José: IBM Corporation, 1995. 477 p.

KAHANI, M.; BEADLE, H. W. P. *Decentralised approaches for network management. Computer Communications Review*. [S.1]. vol. 27,3, p. 36-47, july 1997. (Association for Computing Machinery -ACM).

KLERER, Mark S. *System management information modeling*. **IEEE Communications magazine**, p. 38-43, USA, May 1993.

LABROU, Y.; FININ, T. *A Semantics Approach for KQML: a General Purpose Communication Language for Software Agents*. USA: [s.n], 1996.

LEE, C.; KE, C.; CAHNG, J.; CHEN, Y. *Minimization of Resource Consumption for Multidatabase Query Optimization*. Technical Report. Department of Computer Science, National Cahng-Kung University, Taiwan, 1999.

LEE, K.; PARK, S. *Stable Transaction Management for Preserving the Global Integrity Constraints in Multidatabase Systems*. US: IEEE Computer Society Press, 1998.

MAES, Pattie. *Agents that reduce work and information overload*. Comm. ACM, vol. 37, n. 7, USA, 1994.

MELO, Rubens Silva; S. D. da; TANAKA, A. K. *Banco de Dados em Aplicações Cliente Servidor*. Rio de Janeiro: InfoBook, 1997.

MEYER, K.; ERLINGER, M.; BETSER, J.; SUNSHINE, C.; GOLDSMIDT, G.; YEMINI, Y. *Decentralizind control and intelligence in network management*. In: *International Sympossium on Integrated Network Management, 4th*, 1995, Santa Bárbara, EUA, may de 1995.

MORZY T.; KROLIKOWSKI Z. *Query Optimization in Multidatabase Systems: Solutions and Open Issues*. Proceedings of the 1<sup>st</sup> International Workshop on Database & Expert Systems Applications. US: IEEE Computer Society Press, 1998

MURAKAMI K.; M. KATOH. *Control Architecture for Next-Generation Communication Networks Based on Distributed Databases*. *IEEE Journal Selected Areas in Comm.*, vol. 7, pp. 418 - 423, April 1989.

ORACLE, Inc. *Oracle 7 Concepts Manual*. Estados Unidos: Oracle Corporation, 1996.

\_\_\_\_\_. *Oracle 7 Server Distributed Systems*. vol. I. Distributed Data. Estados Unidos: Oracle Corporation, 1996.

\_\_\_\_\_. \_\_\_\_\_. vol. II. Distributed Data. Estados Unidos: Oracle Corporation, 1996.

\_\_\_\_\_. *Oracle 8 Concepts Manual*. Estados Unidos: Oracle Corporation, 1999.

OZSU, M. Tamer; VALDURIEZ, Patrick. *Principles of Distributed Database Systems*. 2 ed. New Jersey: Prentice Hall, US, 1999.

ROSE, M.T. *The simple book - An introduction to management of TCP/IP based internetworks*. USA: Prentice Hall, 1991.

RUBINSTEIN, M.G. *Avaliação de Desempenho de Agentes Móveis no Gerenciamento de Redes*. 2001. Tese (Doutorado) - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

SÁ, E. S. de. (Coord.). *Manual de normalização de trabalhos técnicos, científicos e culturais*. 6 ed. Rio de Janeiro: Vozes, 2001.



SCOTT, A. Carlisle; CLAYTON, Jan E.; GIBSON L. A. *Practical Guide to Knowledge Acquisition, in Effective Communications*. [s.n]: Addison-Wesley Publishing Company, Inc., 1991. p. 325-327.

SIBERSHCHATZ, Korth; SUDARSCHAN T. **Sistema de Banco de Dados**. São Paulo: Makron Books, 1999.

SINGHAL, S. e ZYDA, M. *Networked virtual environments: design and implementation*. [s.n]: Addison-Wesley Pub, 1999.

SLUMAN, C. *A tutorial on OSI management*. *Computer Network* 17, 4&5 (outubro de 1989), 270-278. Elsevier Science.

STALLINGS, W. *SNMP and SNMPv2: The infrastructure for Network management*. *IEEE Communications Magazine*, New York, vol. 36, n. 03, p.30-45, mar./1998.

\_\_\_\_\_. **SNMP, SNMPv2, SNMPv3 and RMON 1 and 2**. 3th ed. MA: Addison-Wesley Pub, 1999.

STARLING, G.; CARVALHO, A. **Guia inteligente**: Tecnologias de Redes – LAN's, WAN's, Protocolos, Serviços, Modelos e Internet. Petrópolis: Book Express, 1998.

TANENBAUM, Andrew S. **Redes de Computadores**. Rio de Janeiro: Campos, 1997.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. **Pesquisas sobre Bancos Paralelos e Distribuídos**. Apresenta informações sobre Bancos de Dados Paralelos e distribuídos. Disponível em: <<http://www-gppd.inf.ufrgs.br>>. Acesso em: fev. 2002.