

UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

MÁRCIA REGINA DA SILVA PIMENTEL

INVESTIGANDO A RECUPERAÇÃO DE DADOS ESPACIAIS VIA WEB USANDO
OS PRODUTOS ORACLE

MARINGÁ
2005

MÁRCIA REGINA DA SILVA PIMENTEL

INVESTIGANDO A RECUPERAÇÃO DE DADOS ESPACIAIS VIA WEB USANDO
OS PRODUTOS ORACLE

Monografia apresentada à
Universidade Estadual de
Maringá como requisito
parcial para obtenção do
título de Especialista em
Desenvolvimento de
Sistemas para Web.

Orientador
Prof. Dr. Ricardo R. Ciferri
Co-orientador
MSc. Sandro Danilo Gatti

Maringá
2005

MÁRCIA REGINA DA SILVA PIMENTEL

INVESTIGANDO A RECUPERAÇÃO DE DADOS ESPACIAIS VIA WEB USANDO
OS PRODUTOS ORACLE

Monografia apresentada à
Universidade Estadual de
Maringá como requisito
parcial para obtenção do
título de especialista em
Desenvolvimento de
Sistemas para Web.

Aprovado em

BANCA EXAMINADORA

Prof. Wesley Romão
Universidade Estadual de Maringá – UEM

Prof. Marcelo Morandini
Universidade Estadual de Maringá – UEM

Profa. Maria Madalena Dias
Universidade Estadual de Maringá – UEM

Ao meu esposo Cristiano Abram
Pimentel por todo o carinho e dedicaço.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela vida, saúde e inteligência, por me permitir concluir esse trabalho e pelas pessoas maravilhosas que colocou no meu caminho.

Aos meus queridos pais Antonio e Julieta, pelo amor e incentivo, sem os quais eu nunca teria chegado até aqui. Impossível expressar em palavras meu amor e gratidão por vocês!

Ao meu esposo Cristiano, meu incentivador mor, amigo, companheiro, crítico. Esteve em todos os momentos ao meu lado, entendeu todos os momentos de ausência me apoiando em todas as decisões. Você foi à pessoa mais importante nessa etapa da minha vida. Esse trabalho é para você!

Obrigada ao meu orientador, professor Ricardo Rodrigues Ciferri, pelo apoio em todos os momentos, conhecimento passado e, principalmente, pela amizade durante esses meses. Ao meu co-orientador, Sandro Danilo Gatti, pelos conhecimentos em Oracle Spatial transmitidos e paciência em me atender.

Aos meus amigos virtuais Fernando Simon (Univali), André Cruvinel Resende (UFRGS) pela ajuda no MapServer, Daniel Guimarães Almeida (Oracle). Meninos, sem vocês teria sido impossível.

Aos meus amigos da Dental Press, em especial Carlos e Júnior, grandes amigos e incentivadores, Marlene, Marisa e Ademir sempre ao meu lado e sempre nas normas obrigado por todo apoio e ajuda e a querida Mary pela disposição em me ajudar nas traduções.

Aos meus amigos Sérgio, Iza, Fernando, Luiz por todo o apoio dado e a todos os meus amigos que direta ou indiretamente me ajudaram na conclusão desse trabalho. Valeu pessoal!!

"No fim, tudo dá certo. Se não deu, é
porque ainda não chegou ao fim."

Fernando Sabino

Resumo

As características particulares dos dados espaciais são as razões pelas quais se faz necessário estudos sobre o comportamento, a forma de armazenamento e a recuperação desses dados. Este trabalho visa investigar o comportamento de dados espaciais na *World-Wide Web* quanto à forma de armazenamento, a recuperação e a visualização desses dados. Para tanto, foi utilizado o Sistema Gerenciador de Banco de Dados (SGBD) Oracle com o seu módulo para suporte espacial. Para que houvesse a visualização dos dados espaciais foi utilizado o programa MapServer, software de código aberto que permite a visualização de dados espaciais na *World-Wide Web*. Os resultados obtidos nos permitiu concluir que o MapServer é um programa de fácil uso e bastante difundido entre os usuários de SGBD espaciais. O SGBD Oracle, por sua vez, é uma excelente ferramenta para tratar dados espaciais, porém a visualização desses dados é deficiente por se restringir à linguagem SQL. Com relação à indexação, o SGBD Oracle permite usar a R-tree e a Quad-tree, além de permitir a reconstrução de índices.

Palavras-chave: Banco de Dados Espaciais. World-Wide Web. Sistema Gerenciador de Banco de Dados Oracle. Produto Oracle Spatial. Estruturas de Indexação Espacial e Métodos de Acesso. MapServer.

ABSTRACT

The particular characteristics of the spatial data are the reasons that make the studies about the behavior, the form of storage and the recovery of this data necessary. This work aims to investigate the spatial data behavior on the World-Wide Web, considering the form of storage, the recovery and the visualization of these data. For this, the Oracle Database Management System was used with its module for spatial support. The program MapServer, the open source software which permits the visualization of the special data on the World-Wide Web was used in order to visualize the special data. The results obtained allowed us to conclude that the MapServer is a program of easy usage and it is well spread out among the spatial DBMS users. It was confirmed that the Oracle DBMS is an excellent tool for treating special data, however the visualization of these data is deficient, because it is restricted to SQL language. Concerning indexation, the Oracle DBMS allows using the R-tree and the Quad-tree, besides allowing the indices reconstruction.

Key Words: Spatial Database. World-Wide Web. Database Management System. Spatial Oracle Product. Spatial Indexation Structure and Forms of Access. Map Server.

LISTA DE FIGURAS

Figura 1 – Tipos de dados espaciais.....	17
Figura 2 – Tipos de REM.....	19
Figura 3 – Tesselação.....	21
Figura 4 – Modelo de dados do Oracle Spatial.....	25
Figura 5 – Tipos de dados do Oracle Spatial.....	26
Figura 6 – Modelo de consultas.....	33
Figura 7 – Diagrama de aplicação típica do MapServer.....	37
Figura 8 – Área de estudo da aplicação.....	42
Figura 9 – Resultado da consulta.....	49

LISTA DE TABELAS

Tabela 1 - Tipos de elementos espaciais.....	31
Tabela 2 - Operadores e funções espaciais.....	35
Tabela 3 - Tipos de dados MapServ CGI.....	40

LISTA DE ABREVIATURAS - SIGLAS

SGBD	Sistema Gerenciador de Banco de Dados
SGBDE	Sistema Gerenciador de Banco de Dados Espaciais
MapServer	Servidor de Mapa
CAD/CAE	Computer Aided Design/Computer Aided Engineering
REM	Retângulo Envolvente Mínimo
SQL	Structure Query Language
XML	Extensible Markup Language
CGI	Common Gateway Interface
URL	Uniform Resource Locator
OGR	Simple Features Library
GDAL	Geospatial Data Abstraction Library
GML	Geography Markup Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol

SUMÁRIO

1	INTRODUÇÃO	14
1.1	MATERIAIS E MÉTODOS.....	14
1.2	ESTRUTURA DA MONOGRAFIA.....	15
2	BANCO DE DADOS ESPACIAIS	16
2.1	TIPOS DE DADOS.....	16
3	INDEXAÇÃO ESPACIAL	19
3.1	ÍNDICE R-TREE.....	19
3.1.1	Vantagens e Desvantagens do uso da R-tree	20
3.2	ÍNDICE QUAD-TREE.....	20
3.2.1	Vantagens e Desvantagens do uso da Quad-tree	22
4	ORACLE 10g	23
4.1	ORACLE SPATIAL.....	25
4.1.1	Modelo de dados	25
4.1.2	Tipos de Geometria	26
4.1.3	Criação e inserção de elementos na tabela com atributos espaciais e não espaciais	29
4.1.4	Criação de índice espacial	33
4.1.5	Consultas espaciais	34
4.1.6	Operadores e Funções Espaciais do Oracle Spatial	36
4.1.7	Tolerância	37
5	MAPSERVER	38
5.1	ANATOMIA DE UMA APLICAÇÃO MAPSERVER.....	39
5.2	MAPSERV CGI.....	42
5.2.1	Tipos de variáveis	42

5.3	ORACLE SPATIAL COM MAPSERVER.....	43
6	ESTUDO DE CASO	44
6.1	DESCRIÇÃO DOS DADOS CRIADOS NO SGBD ORACLE.....	44
6.2	DESCRIÇÃO DO CÓDIGO MAPSERVER.....	49
7	CONCLUSÕES	52
	REFERÊNCIAS	54

1 INTRODUÇÃO

No atual estágio de desenvolvimento da Internet, a tecnologia permite uma distribuição de informações muito mais interativa e dinâmica do que há alguns anos. Com relação aos dados espaciais, a Internet se apresenta como uma forma de acesso eficiente à informação, porém, o gerenciamento de tais dados é muito mais complexo do que o gerenciamento feito para dados convencionais. Para que haja um desempenho satisfatório é necessário investigar, por exemplo, políticas de distribuição dos dados e estruturas de indexação espacial.

Houve uma grande evolução nos Sistemas Gerenciadores de Banco de Dados (SGBD) convencionais, permitindo o armazenamento, a indexação e a recuperação de dados espaciais, através de uma extensão espacial.

Este trabalho objetiva o estudo de um SGBD espacial, bem como os sistemas de armazenamento, indexação e recuperação dos dados no ambiente world-wide web (ou simplesmente web). Mais especificamente, este trabalho visa apresentar uma análise da extensão espacial do SGBD Oracle por meio de um exemplo em relação à forma de recuperação dos dados espaciais no ambiente web.

1.1 MATERIAIS E MÉTODOS

Para a realização desse trabalho foi necessária a leitura de artigos, teses e dissertações no tema banco de dados espaciais; bem como a leitura de manuais e livros sobre os produtos Oracle e Oracle Spatial; além do estudo da ferramenta MapServer e seus componentes e a realização de testes de desempenho com o produto Oracle Spatial.

1.2 ESTRUTURA DA MONOGRAFIA

Este trabalho está dividido em 7 capítulos, sendo no primeiro uma introdução geral sobre o trabalho e a metodologia utilizada. No segundo capítulo apresentamos alguns conceitos sobre banco de dados espaciais e seus tipos de dados. No capítulo 3 apresenta conceitos sobre indexação espacial e os tipos R-tree e Quad-tree. No capítulo 4 abordamos o tema Oracle 10g descrevendo sua estrutura e o seu módulo espacial, Oracle Spatial. Mostramos os modelos de dados utilizados, tipos de geometria utilizados e como é feita a criação, inserção e consulta de dados espaciais. No capítulo 5 descrevemos a estrutura do MapServer, formato da aplicação e como se dá sua conexão com o Oracle Spatial. No capítulo 6 mostramos um exemplo de dados espaciais na web utilizando MapServer e Oracle Spatial e no capítulo 7 encerramos com nossas conclusões.

2 BANCO DE DADOS ESPACIAIS

Um Sistema Gerenciador de Banco de Dados Espaciais (SGBDE) permite o armazenamento e a manipulação de dados espaciais, que são descritos geometricamente em um espaço bidimensional, tridimensional ou n-dimensional (RIGAUX et al., 2000). Tais sistemas são utilizados em diversas áreas do conhecimento, entre elas aplicações que envolvem imagens de satélite, médicas, ambientes de projeto de engenharia (CAD/CAE) e estruturas moleculares complexas. São fundamentais para o desenvolvimento de aplicações que envolvam planejamento urbano e rural, cadastramento de terras e todo tipo de atividade que envolva o meio ambiente.

Dois pontos importantes de um SGBDE é o volume de dados manipulado e a complexidade dos algoritmos. A área de uma cidade pode ser representada por um polígono com centenas de vértices e um algoritmo que calcule determinada área da cidade onde se concentra determinada população tem uma complexidade significativa.

2.1 TIPOS DE DADOS

Dado espacial é um termo usado para representar fenômenos do mundo real através de duas características básicas:

- Localização geográfica, isto é, sua posição em um sistema de coordenadas conhecido;
- Atributos descritivos.

Um dado espacial pode ser representado por dois modelos de dados distintos que são vetorial ou matricial (*raster*). O modelo de dados vetorial é utilizado

para representar o espaço como um conjunto de entidades discretas definidas por uma unidade que pode ser ponto, linha, polígono, etc.

O modelo de dados matricial é utilizado para representar o espaço geográfico sobre uma superfície contínua, sobre a qual variam os fenômenos a serem observados segundo diferentes distribuições. É representado por uma matriz $M(K \times L)$ formada por K linhas e L colunas, onde cada célula possui a sua localização (um par de coordenadas) e um valor correspondente ao atributo estudado.

Segundo Ciferri (2002) um ponto é a menor unidade possível para representar um objeto espacial. Em geral são usados para representar localizações discretas como uma cidade num mapa. Uma linha pode ser definida como um conjunto de pontos seqüenciais conectados retilineamente. Contrariamente, em uma linha poligonal os pontos não estão conectados de forma retilínea; em ambos os casos, cada par de pontos ligados corresponde a um segmento de linha e são utilizadas para representar objetos espaciais lineares, como rios, estradas, rede elétrica, etc.

Um polígono é formado por uma seqüência de linhas ou linhas poligonais, porém essa seqüência é fechada, isto é, o primeiro ponto coincide com o último. Polígonos e linhas fechadas são equivalentes, diferindo somente em alguns SGBDs espaciais com relação ao armazenamento explícito de atributos convencionais derivados de polígonos, como exemplo área e perímetro (Ciferri, 2002). Polígonos complexos permitem “buracos” ou partes disjuntas. Polígonos e linhas poligonais fechadas são usados para representar objetos bidimensionais como a área ocupada em um bairro.

Um poliedro é limitado por quatro ou mais polígonos, que são designados como face. As interseções das faces formam as arestas e as interseções das arestas formam os vértices. São utilizados para representar sólidos no espaço como um membro humano ou uma peça automotiva.

Os tipos de dados ponto são estruturas 0-dimensional, linhas são 1-dimensional, polígonos são 2-dimensional e poliedros 3-dimensional.

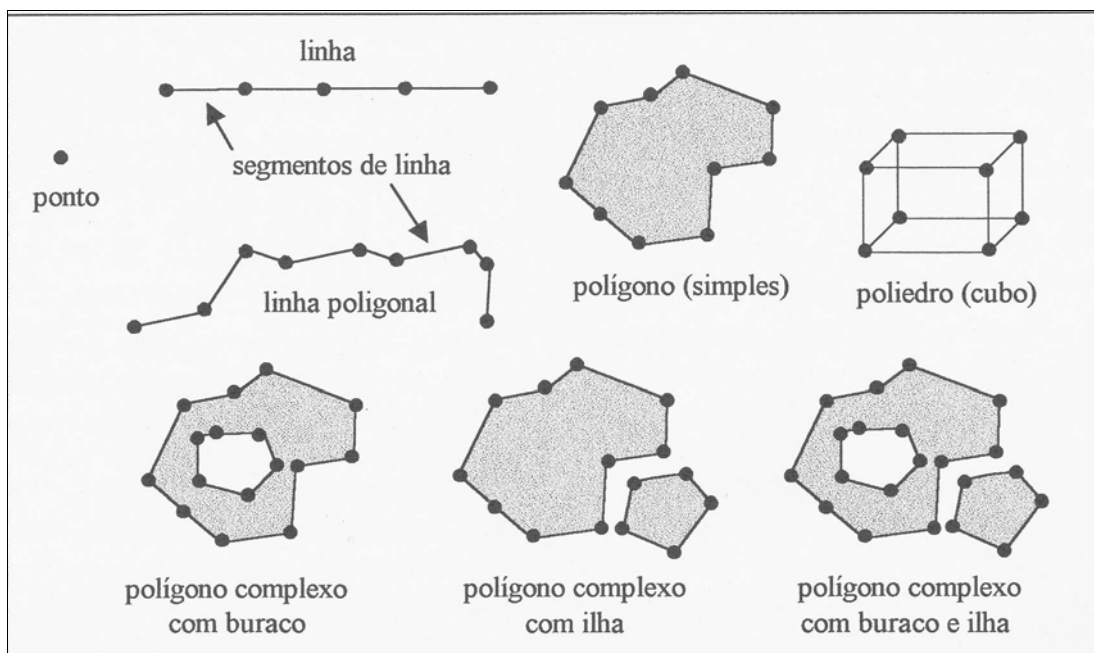


Figura 1 – Tipos de dados espaciais.
Fonte: Ciferri (2002).

3 INDEXAÇÃO ESPACIAL

Índices são estruturas de dados desenvolvidas para otimizar o acesso aos bancos de dados, que, normalmente, são armazenados em memória secundária (Silva, 2002). No caso de banco de dados espaciais, os tipos de estruturas de dados dinâmicas como hash e B-tree não são aplicadas devido a sua natureza multidimensional.

Os índices espaciais levam em consideração a localização do objeto no espaço. Uma grande variedade de métodos de acesso para dados espaciais foi desenvolvida (GATTI, 2000). Neste trabalho estudaremos apenas os índices R-tree e Quad-tree por serem utilizados pela ferramenta Oracle Spatial.

3.1 ÍNDICE R-TREE

Segundo Guttman (1984), uma R-tree é uma estrutura altamente balanceada semelhante a B-tree com índices em cada nó folha contendo ponteiros para os objetos. Cada nó corresponde a uma página de disco no índice residente no disco, e a estrutura é projetada para que a pesquisa espacial necessite visitar apenas um número pequeno de nós. Este índice é completamente dinâmico, fazendo com que inserções e remoções possam ser alternados com a pesquisa dos dados sem que haja reorganização periódica (HOEL e SAMET, 1992).

O índice R-tree é construído utilizando-se apenas uma simples aproximação da geometria do objeto espacial, ou seja, o retângulo envolvente mínimo (REM) à geometria do objeto, podendo ser objetos lineares ou com superfície (GATTI, 2000). Para uma *layer* (i.e., geometrias que compartilham um

conjunto de atributos comuns (Murray, 2002)), um índice R-tree consiste numa estrutura hierárquica dos REMs da geometria na *layer*, como mostra a figura abaixo.

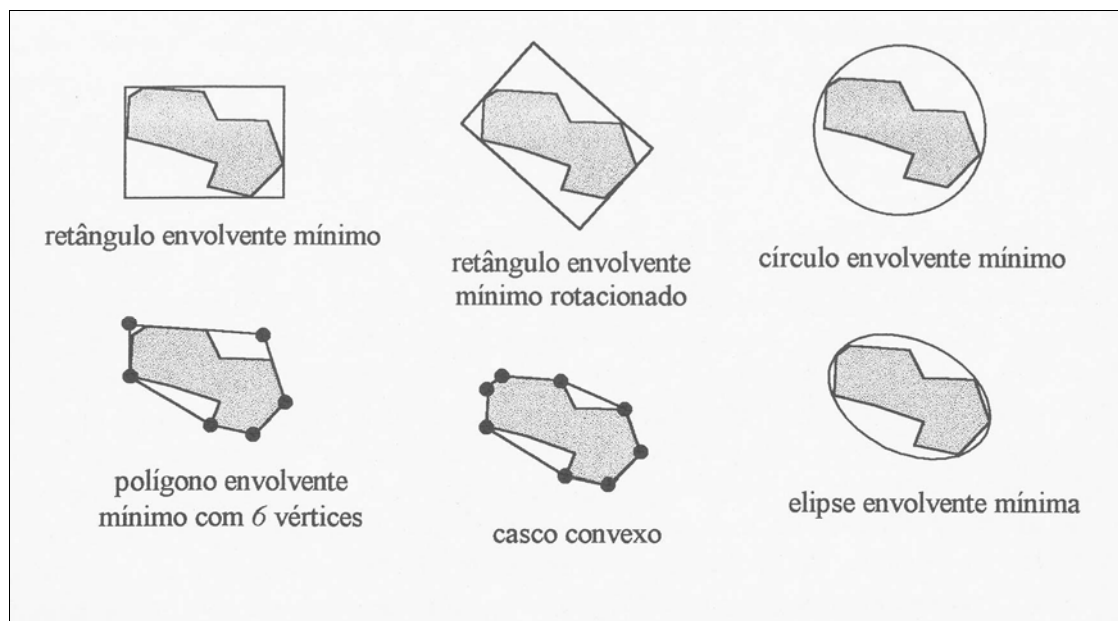


Figura 2 – Tipos de REM.
Fonte: Murray (2002).

3.1.1 Vantagens e Desvantagens do uso da R-tree

Segundo Murray (2002) a R-tree pode indexar dados espaciais de até 4 dimensões no Oracle Spatial. Requer pequena capacidade de armazenamento, são fáceis de serem criados. São mais rápidos em consultas que levam em consideração o vizinho mais próximo, entretanto, não é uma boa escolha se houver atualização constante.

3.2 ÍNDICE QUAD-TREE

Segundo Murray (2002) um índice Quad-tree é um mecanismo de codificação para determinar múltiplas chaves ou seções para cada geometria indexada. Podemos dizer que seções são como retângulos que definem uma cobertura exclusiva (quando nenhuma seção se sobrepõe à outra) e exaustiva

(quando as seções cobrem completamente o objeto) de cada geometria armazenada na *layer*.

Na estrutura Quad-tree, o processo de decomposição do espaço é chamado tesselação. Na tesselação, o espaço de coordenadas é decomposto de maneira hierárquica e regular. No primeiro nível de decomposição, o espaço é dividido em quatro seções. Cada seção que interage com a geometria que está sendo decomposta, é também subdividida em outras quatro seções e assim sucessivamente até que se alcance o limite estipulado de divisão. Os resultados da tesselação são armazenados no índice, uma árvore quaternária cujos nós ou são folhas (não têm filhos) ou têm quatro filhos ordenados e a cada folha pode ser armazenada uma página de disco (GAEDE e OLIVER, 1998).

Murray (2002) diz que o limite de parada para construção do índice pode estar relacionado com a resolução da seção, ou seja, pode-se determinar um tamanho fixo de resolução da seção como único fator de controle. Assim, a tesselação termina quando o espaço de coordenadas pode ser decomposto em um número específico de vezes, tendo cada seção o mesmo tamanho. Segundo o mesmo autor, índices espaciais utilizando seções de tamanho fixo facilitam a comparação entre duas geometrias, que interagem quando compartilham uma ou mais seções.

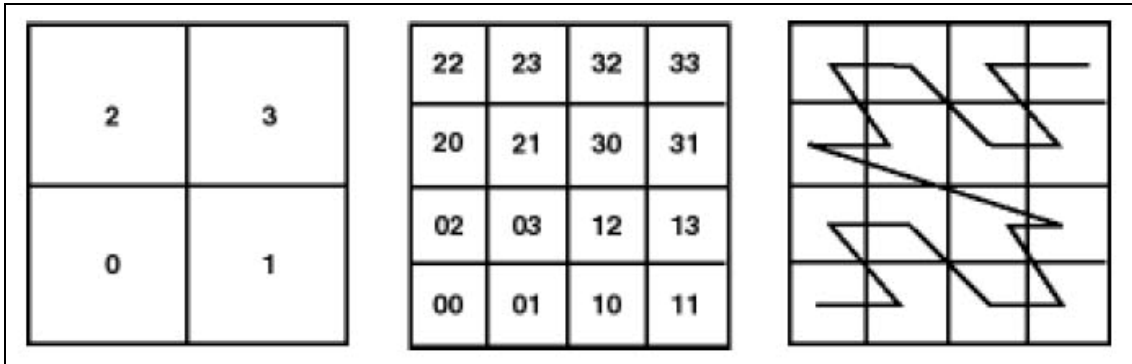


Figura 3 – Tesselação.
Fonte: Murray (2002).

3.2.1 Vantagens e Desvantagens do uso da Quad-tree

Segundo Murray (2002) a eficiência desse método depende do número de níveis e da variação dos tamanhos das geometrias das *layers*. Quando os mesmos tamanhos de seções são utilizados para cobrir geometrias com grande diferenças de tamanho, um grande número de seções será necessário para a geometria maior. Porém, se o tamanho da seção for muito grande, essas seções não representarão geometrias pequenas muito bem. Por isso, a escolha do tamanho da seção traz conseqüências diretas nos resultados das consultas espaciais.

O índice Quad-tree é indicado para os casos em que ocorrem atualizações constantes, pois o índice continua utilizável por bastante tempo mesmo havendo atualizações, sem a necessidade de reconstrução. Em geometrias do tipo polígono ou linha a duplicação das referências dos objetos ocorre em vários níveis, pois podem ocupar mais de uma seção no espaço. Em comparação ao índice R-tree, este necessita maior espaço para armazenamento.

4 ORACLE 10g

O sistema gerenciador de banco de dados Oracle trata os dados de uma única forma. A finalidade deste SGBD é armazenar e recuperar as informações que estão relacionadas. Em geral, um único usuário controla seguramente uma grande quantidade de dados em um ambiente multiusuário, de modo que muitos usuários possam acessar simultaneamente os mesmos dados. Tudo isto é possível em virtude do alto desempenho do SGBD, além de impedir acessos não autorizados e fornecer soluções eficientes para a recuperação de falhas (Cyran, 2003).

O Oracle foi o primeiro SGBD projetado para realizar grandes processamentos de maneira eficaz com um baixo custo para gerenciar informações e aplicações. Com esta arquitetura, cada sistema novo pode ser rapidamente fornecido de um *pool* de componentes. Não há necessidade de picos de *workloads*, porque ao banco pode facilmente alocar ou adicionar recursos conforme a necessidade.

O SGBD Oracle possui tanto estrutura lógica como física. As estruturas físicas e lógicas são separadas, então o armazenamento físico dos dados pode ser controlado sem afetar o acesso às estruturas lógicas do armazenamento (Cyran, 2003).

O SGBD Oracle inclui vários tipos de dados para assegurar todos os tipos comumente utilizados na internet, tais como texto, áudio, vídeo, imagem, XML e dados espaciais. Estes tipos de dados são nativos dentro do SGBD. Todos eles podem ser manipulados utilizando comandos SQL. Um simples comando SQL pode incluir os dados que pertencem a alguns ou a todos estes tipos.

Enquanto as aplicações evoluem para abranger uma semântica cada vez mais rica, ainda encontramos a necessidade de tratar os seguintes tipos dos dados:

- Estrutura de dados simples;
- Estrutura de dados complexas;
- Dados semi-estruturados;
- Dados sem estrutura.

Tradicionalmente, o modelo relacional foi muito bem sucedido em tratar de estruturas de dados simples. A Oracle adicionou características objeto-relacionais de modo que as aplicações pudessem tratar de estruturas de dados complexos, como exemplo coleções, referências, tipos definidos pelo usuário dentre outras.

Os dados não estruturados não podem se decompor em componentes padrão. Os dados sobre um empregado, por sua vez podem 'ser estruturados' em um nome (provavelmente uma cadeia de caracter), em uma identificação (provavelmente um número), em um salário, e assim por diante. Mas se for agregada uma foto, os dados encontrados consistem realmente em uma longa cadeia de 0s e de 1s. Estes 0s e 1s são usados para ligar ou desligar pixels, de modo que quando se vê a foto não é possível quebrá-la como em uma estrutura de dados comum armazenada em uma base de dados. Os tipos de dados utilizados nesta monografia são dados espaciais que fazem parte da extensão espacial deste banco de dados.

4.1 ORACLE SPATIAL

Oracle Spatial é um conjunto integrado de funções e procedimentos que habilita dados espaciais serem armazenados, acessados e manipulados de forma rápida e eficiente no SGBD Oracle. O Oracle Spatial possui arquitetura integrada, implementa o modelo objeto relacional, está presente desde a versão 8i 1.5 e atualmente encontra-se na versão 10g. Integra dados alfanuméricos e espaciais através da criação de novos tipos de dados (MURRAY, 2002).

O Oracle Spatial fornece funções de SQL que facilitem o armazenamento, a recuperação, e a atualização de dados espaciais em uma base de dados que consiste nos seguintes componentes:

- Esquema (MDSYS) que descreve o armazenamento, a sintaxe, e a semântica de tipos de dados geométricos que são permitidos no SGBD Oracle;
- Mecanismo de indexação espacial;
- Conjunto de operadores e funções para executar consultas da área-de-INTERESSE, junções espaciais e outras operações de análise espacial;
- Ferramentas administrativas.

4.1.1 Modelo de dados

O modelo de dados espaciais do Oracle Spatial consiste-se em uma estrutura hierárquica de elementos, geometrias e *layers*, o que corresponde à representação dos dados espaciais. *Layers* são compostas de geometrias que por sua vez são compostas de elementos, como mostra a figura a seguir:

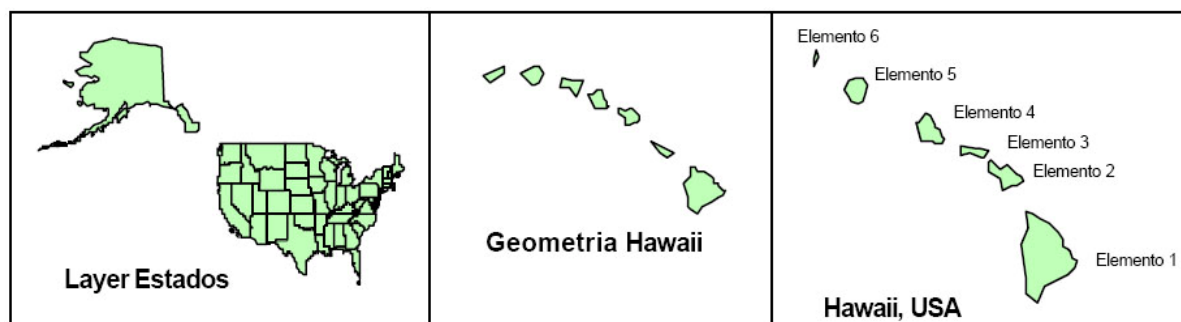


Figura 4 - Modelo de dados do Oracle Spatial.
Fonte: Silva (2002).

- **Elemento:** Componente básico de construção de geometrias, ou seja, são os tipos primitivos de dados disponibilizados pelo Oracle. São construídos utilizando-se coordenadas e, dependendo do tipo de elemento, pode ter um ou vários pares de coordenadas.
- **Geometria:** Representação de um objeto espacial, a partir de elementos primitivos que pode ser uma coleção de elementos homogêneo ou heterogêneo.
- **Layer:** É uma coleção heterogênea de geometrias que compartilham o mesmo conjunto de atributos.

4.1.2 Tipos de Geometria

Dentre os tipos de dados espaciais descritos na seção 2.1, o SGBD Oracle implementa três tipos primitivos de geometria que são ponto, linha e polígono e uma variação de outras geometrias compostas desses tipos primitivos, tais como: arcos circulares, linhas compostas e polígonos conforme mostra a figura 5.

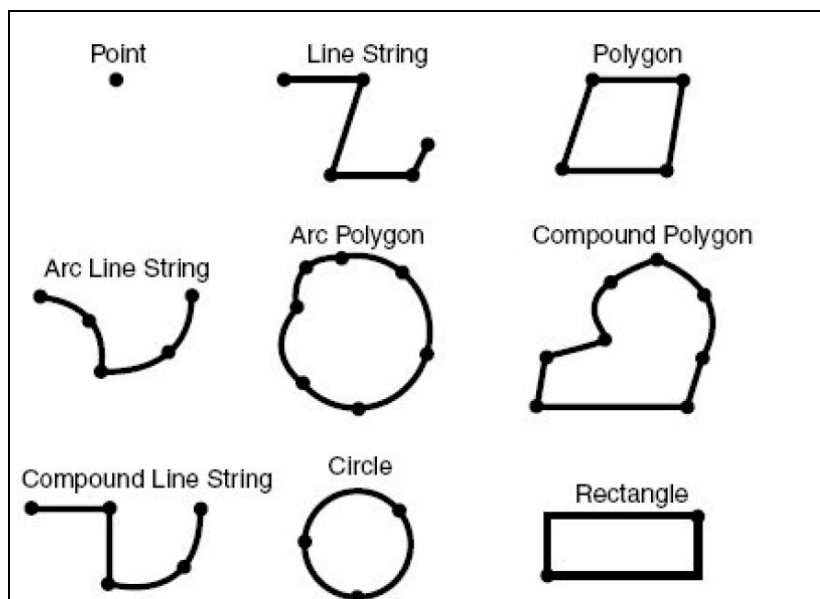


Figura 5 - Tipos de dados do Oracle Spatial.
Fonte: Murray (2002).

Os tipos de dados espaciais bidimensionais são compostos por pontos formados por duas coordenadas X e Y, freqüentemente correspondentes à longitude e latitude. Linhas são compostas de dois ou mais pontos que definem os segmentos de linha. O auto-cruzamento de linhas é permitido, mas o polígono formado no cruzamento não tem um interior implícito; as coordenadas podem ser armazenadas seguindo-se a ordem do sentido horário ou anti-horário.

Polígonos podem conter “buracos” que são construídos, por definição, internos aos polígonos. Neste caso, o anel exterior e o anel interior do polígono são considerados dois elementos distintos que juntos formam um polígono complexo. Ambos os tipos, exterior e interior, devem ser armazenados seguindo a ordem do sentido horário. O auto-cruzamento de polígonos não é permitido, embora o de linha seja. A extensão também permite o armazenamento e a indexação de tipos tridimensionais e tetradimensionais, mas as funções e operadores só funcionam para os tipos bidimensionais.

Baseado no modelo objeto-relacional, o Oracle Spatial define um tipo de objeto, para representar dados espaciais, chamado SDO_GEOMETRY, como mostrado a seguir (MURRAY, 2002).

```
CREATE TYPE sdo_geometry AS OBJECT (  
  SDO_GTYPE NUMBER,  
  SDO_SRID NUMBER,  
  SDO_POINT SDO_POINT_TYPE,  
  SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,  
  SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

Este objeto contém a geometria em si, as suas coordenadas, e informações sobre seu tipo e projeção. Em uma tabela espacial, os atributos alfanuméricos da geometria são definidos como colunas de tipos básicos (VARCHAR2, NUMBER, DATE, dentre outros), e a geometria como uma coluna do tipo SDO_GEOMETRY. Em uma tabela espacial, cada instância do dado espacial é armazenada em uma linha, e o conjunto de todas as instâncias dessa tabela forma um plano de informação. O objeto SDO_GEOMETRY é composto pelos seguintes atributos:

- ❖ SDO_GTYPE: formado por quatro números, sendo que os dois primeiros indicam a dimensão da geometria e os outros dois o seu tipo. Os tipos podem ser: 00 (não conhecido), 01 (ponto), 02 (linha ou curva), 03 (polígono), 04 (coleção), 05 (multipontos), 06 (multilinhas) e 07 (multipolígonos);
- ❖ SDO_SRID: utilizado para identificar o sistema de coordenadas, ou sistema de referência espacial, associado à geometria;
- ❖ SDO_POINT: é definido utilizando um objeto do tipo SDO_POINT_TYPE, que contém os atributos X, Y e Z para representar as coordenadas de um ponto.

Somente é preenchido se a geometria for do tipo ponto, ou seja, se os dois últimos números do SDO_GTYPE forem iguais a "01";

❖ SDO_ELEM_INFO: é um vetor de tamanho variável que armazena as características dos elementos que compõem a geometria. As coordenadas de cada elemento são armazenadas em um vetor variável chamado SDO_ORDINATES e são interpretadas através de três números armazenados no SDO_ELEM_INFO:

- ❖ SDO_STARTING_OFFSET: indica qual a posição da primeira coordenada do elemento no SDO_ORDINATES;
 - ❖ SDO_ETYPE: indica o tipo do elemento;
 - ❖ SDO_INTERPRETATION: indica como o elemento deve ser interpretado juntamente com o SDO_ETYPE.
- ❖ SDO_ORDINATES: é um vetor de tamanho variável que armazena os valores das coordenadas da geometria.

Apresentaremos abaixo a criação de tabelas com campos espaciais, seus atributos e geometria bem como a inserção de elementos nessas tabelas.

4.1.3 Criação e inserção de elementos na tabela com atributos espaciais e não espaciais

Na criação da tabela é necessário criar um campo do tipo espacial. No momento da inserção é que especificamos o tipo da geometria a ser inserida.

```

create table Estados_do_Brasil
{
nome_estado VARCHAR2(100),
nome_capital VARCHAR2(100),
fronteira MDSYS.SDO_GEOMETRY
}

```

Inserção de elemento do tipo retângulo

```

INSERT INTO Estados_do_Brasil VALUES (
'Paraná',
'Curitiba'
SDO_GEOMETRY (
2003,
NULL,
NULL,
SDO_ELEM_INFO_ARRAY(1,1003,3),
SDO_ORDINATE_ARRAY(1,1, 5,7)
));

```

Inserção de elemento do tipo polígono

```

INSERT INTO Estados_do_Brasil VALUES (
'Santa Catarina',
'Florianópolis',
SDO_GEOMETRY (
2003,
NULL,
NULL,
SDO_ELEM_INFO_ARRAY(1,1003,1),
SDO_ORDINATE_ARRAY(3,3, 6,3, 6,5, 4,5, 3,3)
)
);

```

Inserção de elemento do tipo círculo

```
INSERT INTO Estados_do_Brasil VALUES (
  'Bahia',
  'Salvador',
  SDO_GEOMETRY (
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY (1, 1003, 4),
    SDO_ORDINATE_ARRAY (8, 7, 10, 9, 8, 11)
  )
);
```

Inserção de elemento do tipo ponto

```
INSERT INTO cola_markets VALUES (
  90,
  'point_only',
  SDO_GEOMETRY (
    2001,
    NULL,
    SDO_POINT_TYPE (12, 14, NULL),
    NULL,
    NULL)) ;
```

No momento que inserimos os dados espaciais na tabela é que vamos especificar qual o seu tipo e parâmetros necessários. Para isso é necessário montar o objeto SDO_GEOMETRY. O primeiro atributo utilizado é o SDO_GTYPE formado por quatro dígitos na forma dl_{tt}, onde:

d indica a dimensão do atributo;

l é utilizado somente para elementos com 3 ou 4 dimensões. Indica sistema de referência linear;

tt indica o tipo da geometria. Os parâmetros utilizados vão de 00 até 07.

Através do atributo SDO_GTYPE podemos identificar qual o tipo de geometria é inserida no banco de dados, além de informar qual o formato da geometria. Ver valores correspondentes na tabela 1.

Tabela 1 – Tipos de elementos espaciais.

<i>SDO_ETYPE</i>	<i>SDO_INTERPRETATION</i>	<i>Descrição do tipo</i>
0	Nenhum valor	Tipo não disponibilizado
1	1	Ponto
1	n > 1	Conjunto de n pontos
2	1	Linha formada por vértices conectados por segmentos retos
2	2	Linha formada por vértices conectados por arcos circulares
1003 ou 2003	1	Polígono simples composto por vértices conectados por segmentos retos
1003 ou 2003	2	Polígono simples composto por vértices conectados por arcos circulares
1003 ou 2003	3	Retângulo otimizado composto por dois pontos
1003 ou 2003	4	Círculo formado por três pontos da circunferência
4	n > 1	Linha composta por alguns vértices conectados por segmentos de reta e outros por arcos
1005 ou 2005	n > 1	Polígono composto por alguns vértices conectados por segmentos de reta e outros por arcos

O próximo atributo é o SDO_SRID, utilizado para identificar o sistema de coordenada geográfica associado com a geometria. Caso o SDO_SRID seja null, nenhum sistema de coordenadas está associado com a geometria. Em seguida apresentamos o atributo SDO_POINT, definido pelo tipo de objeto SDO_POINT_TYPE, que possui atributos X, Y e Z. Caso o SDO_ELEM_INFO e o

SDO_ORDINATES são ambos nulos e o atributo do SDO_POINT é não nulo, então os valores de X e Y são considerados coordenadas de um ponto.

Os dois últimos atributos são o SDO_ELEM_INFO_ARRAY e o SDO_ORDINATE_ARRAY. O primeiro é definido usando um array de números. Este atributo informa ao banco como interpretar os elementos que estão no atributo SDO_ORDINATES. O segundo é também um array de números que armazena os valores das coordenadas para “montar” o objeto espacial. Estes valores devem ser sempre utilizados conjuntamente com os valores do SDO_ELEM_INFO. Os valores no array são dispostos pela dimensão. Se o objeto for bidimensional os pontos serão armazenados da forma {X1,Y1, X2,Y2, X3,Y3, X4,Y4,X1,Y1}.

Para a inserção de polígonos, em geral, é utilizado a mesma estrutura de insert. Porém quando a inserção é para elemento do tipo ponto a estrutura se altera, ou seja, o SDO_POINT_TYPE possui valores e os atributos SDO_ELEM_INFO e o SDO_ORDINATE não são requeridos.

4.1.4 Criação de índice espacial

A indexação espacial, como qualquer outro tipo de indexação, fornece mecanismos para limitar o conjunto de busca, aumentando assim a desempenho no processamento das consultas para a recuperação de dados espaciais.

```
CREATE INDEX cola_spatial_idx
ON cola_markets(shape)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
-- Criação de índice R-TREE
-- Abaixo especificação para índice Quad-Tree
-- PARAMETERS('SDO_LEVEL = 8');
```

Por padrão, o Oracle Spatial cria o índice sempre como R-tree. Para que a base de dados seja indexada pelo índice quad-tree é necessário adicionar o parâmetro PARAMETERS ('SDO_LEVEL = X') no caso do level é associado um número 0, 1 ou maior que 1 que indica o número de divisões do índice.

4.1.5 Consultas espaciais

O Oracle Spatial utiliza um modelo de consulta baseado em duas etapas, chamadas de primeiro e segundo filtro, como mostrado na figura 6. O primeiro filtro considera as aproximações das geometrias, pelo critério do retângulo envolvente mínimo, para reduzir a complexidade computacional. Este filtro é de baixo custo computacional e seleciona um subconjunto menor de geometrias candidatas que será passado para o segundo filtro. O segundo filtro trabalha com as geometrias exatas, por isso é computacionalmente mais caro e só é aplicado ao subconjunto resultante do primeiro filtro retorna o resultado exato da consulta. Para executar consultas e operações espaciais, o Oracle Spatial fornece um conjunto de operadores e funções que são utilizados juntamente com a linguagem SQL (Murray, 2002).

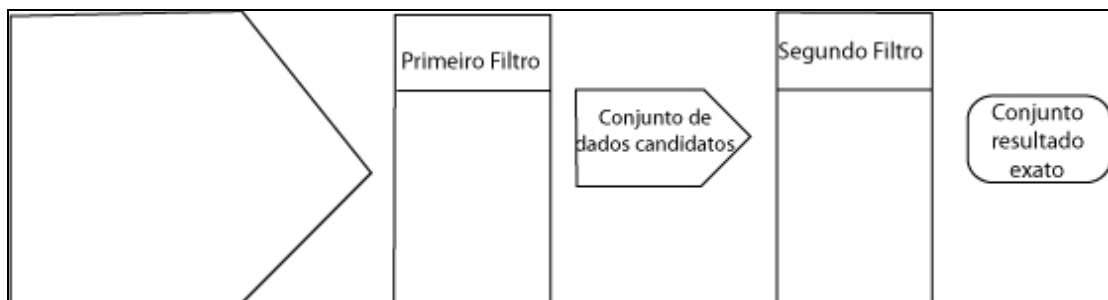


Figura 6 – Modelo de consultas.
Fonte: Murray (2002).

Caso 1: Selecionar todas as cidades que estão ao redor de um estado em particular

```
select county, c.state_abrv
from counties c, states s
where s.state = 'FLORIDA'
and mdsys.sdo_filter (c.geom, s.geom, 'querytype=JOIN') = 'TRUE';
```

Caso 2: Selecione todas as cidades com uma área em particular

```
select c.city, c.pop90
from cities c
where mdsys.sdo_relate (
c.location,
mdsys.sdo_geometry (3, null, null,
mdsys.sdo_elem_info_array (1,3,3),
mdsys.sdo_ordinate_array (-109, 37, -102, 40)),
'mask=ANYINTERACT querytype=WINDOW LAYER_GTYPE=POINT') = 'TRUE';
```

4.1.6 Operadores e Funções Espaciais do Oracle Spatial

Mostramos na tabela abaixo os operadores e funções espaciais permitidos pelo Oracle Spatial.

Tabela 2 - Operadores e funções espaciais.

Operadores Espaciais	
SDO_NN	Determina os vizinhos mais próximos a uma geometria
SDO_NN_DISTANCE	Determina quão distante estão os objetos retornados pelo operador SDO_NN de uma dada geometria
SDO_RELATE	Determina se duas geometrias se interagem (ou não) de algum modo
SDO_WITHIN_DISTANCE	Determina se uma geometria está a uma dada distância de outra
Funções Espaciais	
SDO_GEOM.RELATE	Determina como duas geometrias se interagem
SDO_GEOM.SDO_AREA	Calcula a área de um polígono de duas dimensões
SDO_GEOM.SDO_BUFFER	Gera um buffer (em forma de polígono) ao redor de uma geometria
SDO_GEOM.SDO_DIFFERENCE	Retorna a geometria correspondente à diferença topológica entre duas geometrias
SDO_GEOM.SDO_DISTANCE	Calcula a distância entre duas geometrias
SDO_GEOM.SDO_INTERSECTION	Retorna a geometria correspondente à interseção topológica entre duas geometrias
SDO_GEOM.SDO_LENGTH	Calcula o comprimento ou perímetro de uma geometria
SDO_GEOM.SDO_UNION	Retorna a geometria correspondente à união topológica entre duas geometrias
SDO_GEOM.VALIDATE_GEOMETRY	Determina se uma geometria é válida
SDO_GEOM.VALIDATE_LAYER	Determina se todas as geometrias armazenadas em uma coluna espacial são válidas
SDO_GEOM.WITHIN_DISTANCE	Determina se uma geometria está a uma distância específica (distância Euclidiana) de outra

4.1.7 Tolerância

Tolerância é utilizada para associar uma margem de erro na consulta de dados espaciais. O valor utilizado deve ser um número não negativo e maior que zero. A significância do valor depende se o dado espacial está associado ou não ao sistema de coordenadas geodésicas. Por exemplo, caso queira recuperar todas as farmácias que estão a 100 metros de um determinado ponto com uma tolerância de 0.05m; se uma farmácia estiver a 99.95m ou 100.05m também será recuperada.

- Para dados geodésicos, que são identificados pela latitude e longitude, o valor utilizado para a tolerância é em metros.
- Para dados não geodésicos, o valor utilizado está associado ao sistema de coordenadas associada àqueles dados. Por exemplo, se a unidade de medida for em milhas, uma tolerância de valor dois indica que esta é de duas milhas.

Em ambos os casos, quanto menor o valor, maior precisão está associada com os dados.

5 MAPSERVER

Em sua forma mais básica, o MapServer é um programa CGI que é carregado em seu servidor web. Quando um pedido é emitido ao MapServer, usa-se a informação passada pela URL, um arquivo de mapa é criado e uma imagem é solicitada. A solicitação pode também retornar imagens para legendas, barras de escala, mapas de referência e valores passados como variáveis do CGI. Abaixo se encontra um diagrama de aplicação típica do MapServer (MAPSERVER NEW USERS, 2005).

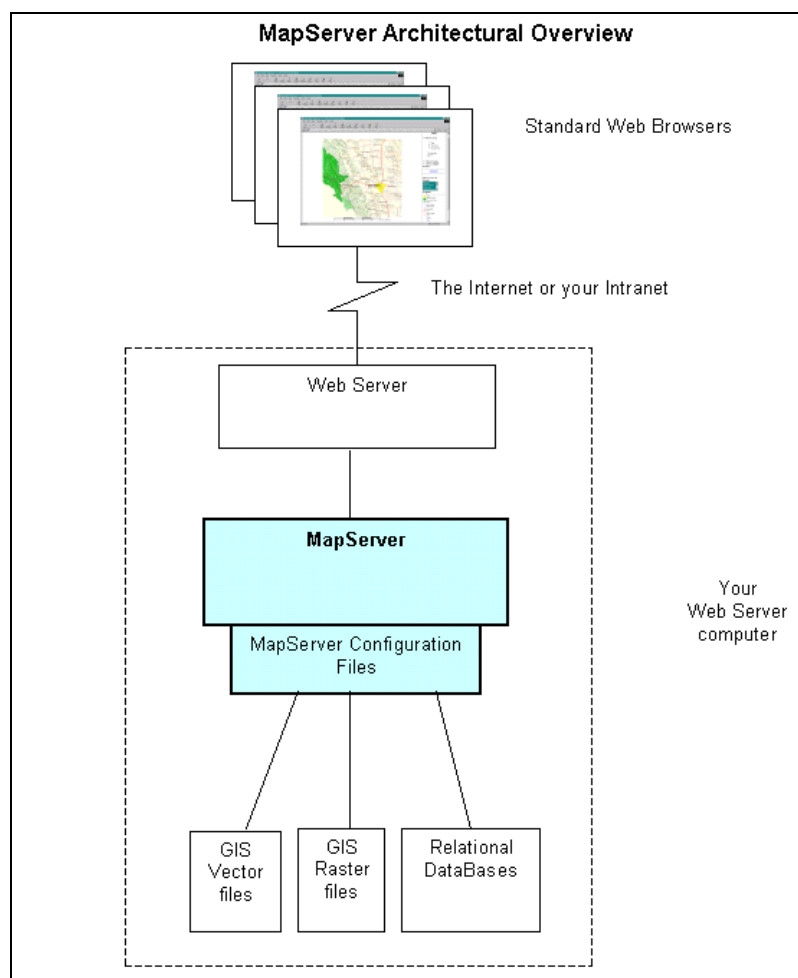


Figura 7 – Diagrama de aplicação típica do MapServer.

Fonte: http://ms.gis.umn.edu/new_users/msappdiagram/image_view_fullscreen.

O MapServer pode ser estendido e customizado. Permite muitos formatos diferentes de entrada e saída. Isto é possível compilando-se o binário MapServer. Muitos dos formatos que ainda não são desenvolvidos para MapServer, podem ser acessados via OGR.

5.1 ANATOMIA DE UMA APLICAÇÃO MAPSERVER

Uma aplicação simples de MapServer consiste de:

- Arquivo Map - é um arquivo de configuração de texto estruturado para uma aplicação MapServer. Define a área do mapa, mostrar ao MapServer onde seus dados estão e onde será a saída das imagens. Define também as camadas do mapa (*layers*), incluindo a origem dos dados, projeções e simbologia. Isto usualmente é usado com a extensão map.
- Dados geográficos - MapServer pode utilizar vários tipos de dados geográficos. O formato *default* é o ESRI *shapefile*. Um ESRI *shapefile* consiste em um arquivo principal, um arquivo de índice, e em uma tabela de banco de dados. O acesso direto é pelo arquivo principal, onde cada registro descreve uma forma com uma lista dos vértices. No arquivo de índice, cada registro contém o valor do registro correspondente ao arquivo principal. A tabela do banco de dados contém atributos de cada registro. O relacionamento um-para-um entre a geometria e os atributos é baseado no número de registros. Os registros do atributo no banco de dados devem estar na mesma ordem que registros no arquivo principal. Muitos outros formatos podem ser permitidos nativamente ou se compilados com as bibliotecas *open source* GDAL e OGR. Tais dados podem ser PostgreSQL-PostGIS, Geography Markup Language (GML), MapInfo entre outros.

- Páginas HTML - é a interface entre o usuário e o MapServer. Normalmente encontra-se na raiz do servidor web, que é um formulário simples, o MapServer pode ser chamado para mostrar uma imagem estática de mapa na página html. Para fazer mapas interativos, a imagem é colocada num formulário html.

Programas CGI são *stateless*, ou seja, cada pedido é novo e não é lembrado o que foi realizado na sua aplicação anterior. Por esta razão, a cada novo pedido da aplicação enviado ao MapServer, é necessário passar o contexto da informação (qual *layer* está, onde está no mapa, modalidade de aplicação, etc.) em variáveis escondidas no formulário ou através de URLs.

Uma simples aplicação deve incluir duas páginas html:

- Arquivo de inicialização - usa um formulário com variáveis escondidas para enviar uma pergunta inicial ao servidor http e ao MapServer. Este formulário pode ser colocado em uma outra página ou ser substituído passando a informação da iniciação como variáveis em uma URL.
- Arquivo *template* – controla como os mapas e legendas serão apresentados no *browser*. Para referenciar variáveis do MapServer CGI na *template* HTML, é necessário permitir que o MapServer faça o povoamento com valores relacionados ao estado atual de sua aplicação (por exemplo nome da imagem do mapa, nome da imagem da referência, extensão do mapa, etc.). A partir disso é criada a página HTML para que o *browser* leia. A *template* determina também como o usuário pode interagir com a aplicação de MapServer (*browse*, *zoom*, *consulta*).
- MapServer CGI - É um arquivo binário ou executável que recebe os pedidos e retorna imagens, dados, etc. Deve ser colocado no diretório de scripts ou no

cgi-bin do servidor http. O usuário do servidor web deve executar privilégios para este diretório, por razões de segurança, não deve estar na raiz do servidor web.

- Servidor HTTP - é necessário um servidor HTTP (web) trabalhando, com o Apache ou o Microsoft Internet Information Server (IIS), na máquina na qual irá instalar o MapServer.

5.2 MAPSERV CGI

MapServ faz parte do pacote MapServer que dá suporte ao CGI. Ele utiliza entradas na criação ou seleção de chamadas. Aceita entradas tanto pelo método “POST” como “GET” usado de uma maneira interativa com as imagens (CGI, 2005).

Informações gerais:

- As variáveis não são *case sensitive*;
- Em casos onde múltiplos valores são associados com uma variável (ex: Mapext), o valor deve ser separado por espaços;
- Os conteúdos de variáveis são checados por tipos de dados apropriados e armazenados;
- Qualquer variável utilizada que não seja definida é armazenada e pode ser referenciada através de uma *template*.

5.2.1 Tipos de variáveis

O MapServer CGI aceita inúmeros tipos de variáveis já definidas.

Descreveremos algumas delas na tabela abaixo.

Tabela 3 – Alguns tipos de dados MapServ CGI.

<i>Tipo</i>	<i>Descrição</i>
IMG	O nome é associado com a imagem do mapa usado pelo usuário. O que atualmente é passado são duas variáveis, img.x e img.y. Para aplicações CGI esta é uma imagem essencial.
Layer	O nome de uma <i>layer</i> como aparece no arquivo do mapa. Enviar ao Mapserv o nome de uma <i>layer</i> ativa o status dessa <i>layer</i> para <i>ativado</i> .
Map	Onde o mapa criado é armazenado. Usado com uma tag .
Reference	Onde o resultado do mapa criado é retornado. Usado com uma tag .
Scale	Escala é utilizada para criar um mapa novo. Usado com mapxy. A escala é dada como o denominador da fração real da escala, por exemplo, para um mapa com uma escala de 1:24000, usa-se 24000.

5.3 Oracle Spatial com MapServer

Oracle Spatial e MapServer podem ser combinados de duas formas:

- Utilizando o suporte nativo ao Oracle Spatial do próprio MapServer; ou
- Através de OGR (USING ORACLE SPATIAL DATA WITH MAPSERVER, 2005).

OGR é uma biblioteca de código aberto que provê leitura; algumas vezes escrita e acesso a uma variedade de formatos de arquivos vetoriais incluindo ESRI Shapefiles, PostGis, Oracle Spatial, entre outros (OGR SIMPLE FEATURE LIBRARY, 2005).

O *driver* OGR normalmente não é utilizado no Oracle Spatial, porém algumas vezes é utilizado em plataformas onde as bibliotecas do Oracle *Client* estão disponíveis. Segundo Simon (2005) o suporte nativo ao Oracle Spatial é quatro vezes mais rápido que via OGR.

Para se conectar ao SGBD Oracle através do MapServer é necessário utilizar a seguinte forma (ORACLE SPATIAL, 2005):

“OCI: userid/password@database_instance: table, table”, onde:

A lista de tabelas é opcional, a porção database_instance pode ser omitida quando acessado de forma local.

6 ESTUDO DE CASO

Para justificar os conceitos apresentados anteriormente, faremos uma pequena demonstração do comportamento de diversos tipos de dados espaciais. O exemplo utilizado é o proposto por Murray (2002) no Oracle Spatial User's Guide and Reference:

6.1 DESCRIÇÃO DOS DADOS CRIADOS NO SGBD ORACLE

Uma empresa de refrigerantes do tipo cola quer identificar áreas estratégicas para a venda de seus produtos. Veja área na figura abaixo:

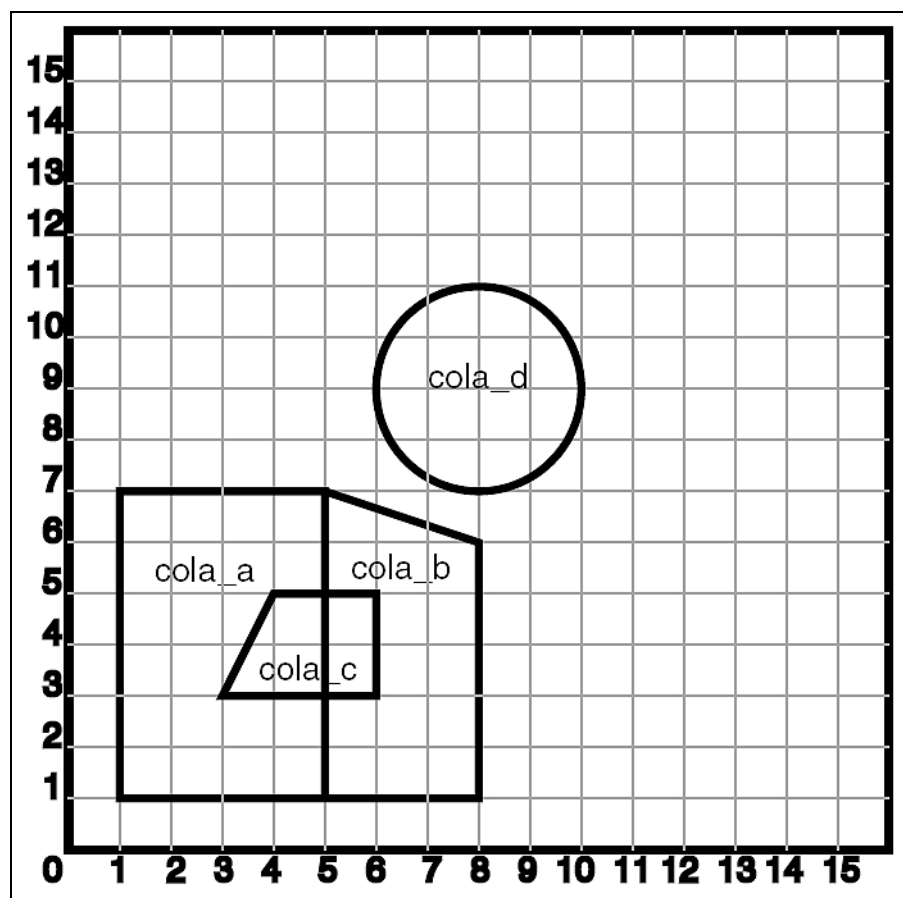


Figura 8 – Área de estudo da aplicação.
Fonte: Murray 2002.

A seguir estão descritos os comandos SQL para a criação do Banco de Dados espacial no SGBD Oracle, descrito em Murray (2002) com a inclusão dos dados.

Primeiramente vamos criar um usuário:

```
CREATE USER MARCIA IDENTIFIED BY marcia.
```

Vamos atribuir direitos a este usuário:

```
GRANT CONNECT TO MARCIA;  
GRANT DBA TO MARCIA;  
GRANT RESOURCE TO MARCIA;  
GRANT EXP_FULL_DATABASE, IMP_FULL_DATABASE TO MARCIA;
```

Criação das tabelas, inserção dos elementos e criação de índice

```
CREATE TABLE cola_markets (  
  mkt_id NUMBER PRIMARY KEY,  
  name VARCHAR2(32),  
  shape SDO_GEOMETRY);  
  
INSERT INTO cola_markets VALUES(  
  1,  
  'cola_a',  
  SDO_GEOMETRY(  
    2003,  
    NULL,  
    NULL,  
    SDO_ELEM_INFO_ARRAY(1,1003,3),  
    SDO_ORDINATE_ARRAY(1,1, 5,7)  
  )  
);
```

```
INSERT INTO cola_markets VALUES(
  2,
  'cola_b',
  SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
  SDO_ORDINATE_ARRAY(5,1, 8,1, 8,6, 5,7, 5,1)
  )
);
```

```
INSERT INTO cola_markets VALUES(
  3,
  'cola_c',
  SDO_GEOMETRY(
  2003, -- two-dimensional polygon
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,1),
  SDO_ORDINATE_ARRAY(3,3, 6,3, 6,5, 4,5, 3,3)
  )
);
```

```
INSERT INTO cola_markets VALUES(
  4,
  'cola_d',
  SDO_GEOMETRY(
  2003,
  NULL,
  NULL,
  SDO_ELEM_INFO_ARRAY(1,1003,4),
  SDO_ORDINATE_ARRAY(8,7, 10,9, 8,11)
  )
);
```

```
INSERT INTO cola_markets VALUES(  
10,  
'polygon_with_hole',  
MDSYS.SDO_GEOMETRY(  
2003, -- 2-dimensional polygon  
NULL,  
NULL,  
MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1, 19,2003,1), -- polygon with hole  
MDSYS.SDO_ORDINATE_ARRAY(2,4, 4,3, 10,3, 13,5, 13,9, 11,13, 5,13, 2,11,  
2,4,  
7,5, 7,10, 10,10, 10,5, 7,5)  
)  
);
```

```
INSERT INTO cola_markets VALUES(  
11,  
'compound_line_string',  
MDSYS.SDO_GEOMETRY(  
2002,  
NULL,  
NULL,  
MDSYS.SDO_ELEM_INFO_ARRAY(1,4,2, 1,2,1, 3,2,2), -- compound line string  
MDSYS.SDO_ORDINATE_ARRAY(10,10, 10,14, 6,10, 14,10)  
)  
);
```

```

INSERT INTO cola_markets VALUES(
12,
'compound_polygon',
MDSYS.SDO_GEOMETRY(
2003, -- 2-dimensional polygon
NULL,
NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,1005,2, 1,2,1, 5,2,2), -- compound polygon
MDSYS.SDO_ORDINATE_ARRAY(6,10, 10,1, 14,10, 10,14, 6,10)
)
);

```

```

INSERT INTO cola_markets VALUES(
90,
'point_only',
MDSYS.SDO_GEOMETRY(
2001,
NULL,
MDSYS.SDO_POINT_TYPE(12, 14, NULL),
NULL,
NULL));

```

```

INSERT INTO cola_markets VALUES(
13,
'type_zero_element_geom',
MDSYS.SDO_GEOMETRY(
2003, -- 2-dimensional polygon
NULL,
NULL,
MDSYS.SDO_ELEM_INFO_ARRAY(1,0,57, 11,1003,3), -- 1st is type 0 element
MDSYS.SDO_ORDINATE_ARRAY(6,6, 12,6, 9,8, 6,10, 12,10, 6,4, 12,12)
)
);

```

```

INSERT INTO USER_SDO_GEOM_METADATA
VALUES (
'cola_markets',
'shape',
MDSYS.SDO_DIM_ARRAY( -- 20X20 grid
MDSYS.SDO_DIM_ELEMENT('X', 0, 20, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y', 0, 20, 0.005)
),
NULL );

CREATE INDEX cola_spatial_idx
ON cola_markets(shape)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

```

6.2 DESCRIÇÃO DO CÓDIGO MAPSERVER

```

map
extent 0 0 20 20
units dd

interlace on
size 400 400
imagecolor 70 201 229
shapepath 'data'
fontset '/home/apache/mapserv/maporaclespatial/fontes/fontes.txt'

symbol
name 'single_point'
type vector
filled on
points
0 0
1 0
1 1
0 1
0 0
end
end

web
imagepath '../..//monografia/tmp/'
imageurl '/tmp/'
end

reference
status on
color -1 -1 -1
extent 0 0 20 20
size 120 109
image '../..//htdocs/monografia/imagens/referencia.png'
outlinecolor -1 -1 -1
end

scalebar
size 400 4
intervals 7

```

```

style 0
units kilometers
color 0 0 0
outlinecolor 0 0 0
label
color 0 0 0
size tiny
end
end

```

Nesta primeira parte do código é definido o tamanho da área mostrada no *browser*, através do *size*, cor apresentada através do *imagecolor*. *Shapepath* é nome da variável que recebe o resultado da consulta e *fontset* é o caminho de onde é armazenado o arquivo de fontes utilizados no script.

A definição de *symbol* foi escolhida para uma visualização melhor do ponto no *browser*, pois o ponto que o MapServer plota na tela é muito pequeno. Na definição *web* esta é utilizada para informar qual o caminho que a imagem gerada pelo MapServer será salva. O item *reference* é utilizado para armazenar uma imagem da área visualizada. Já em *Scalebar* o mais importante é definir cores de linha, unidade de medida utilizada, espessura da linha, entre outros.

```

layer
connectiontype ORACLESPATIAL
connection 'user/pass@sid'
data "shape from (SELECT shape FROM cola_markets WHERE mkt_id = 10) "
name 'polygon_hole'
status off
type polygon
class
outlinecolor 200 0 0
end
end

```

Após a parte de definição, é mostrado como é definido o acesso ao SGBD Oracle. Informa-se o tipo de conexão definindo-se o banco de dados, estabelece-se a conexão informando usuário e senha. A seguir a variável *data* recebe a consulta, a seguir informa-se o nome o qual será chamado à consulta, seu tipo espacial, no caso acima, polígono e a cor da linha usada. Na figura 9 é mostrado o resultado da consulta.

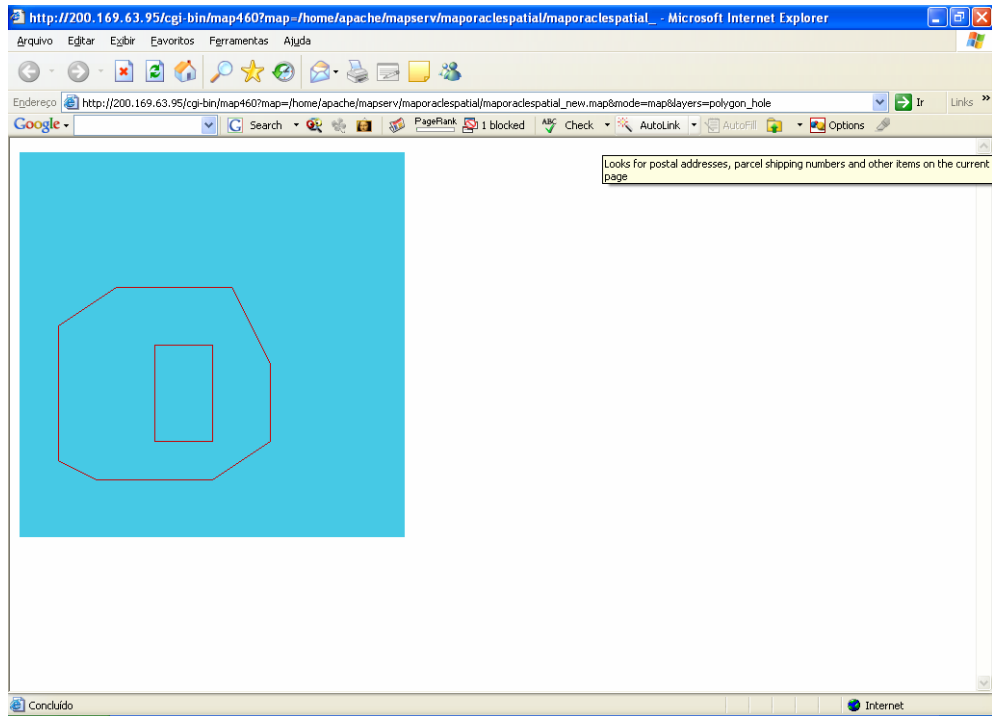


Figura 9 – Resultado da consulta.

7 CONCLUSÕES

Este trabalho abordou os conceitos fundamentais dos SGBDs espaciais principalmente no que se refere aos tipos de dados e operadores espaciais, tipos de consultas, indexação espacial e sua aplicação nas ferramentas Oracle.

A partir disso verificamos que o SGBD Oracle, no seu módulo espacial, é uma excelente ferramenta para tratar de dados espaciais, pois são mantidos da mesma forma que dados não espaciais. Por outro lado, verificamos que a visualização dos dados é deficiente por se restringir à linguagem SQL. Com relação aos tipos de dados é permitido armazenar apenas dados vetoriais.

Nos testes práticos, utilizamos a ferramenta de código aberto MapServer juntamente com a interface CGI para disponibilizarmos os dados na web; este se mostrou bastante eficiente e de rápida visualização. Uma das vantagens de se ter o dado visualizado na web é poder ver as informações de forma individualizada e combinada. A visualização das partes facilita o estudo pontual e/ou de todo o fenômeno. Dependendo da quantidade de informações que se tem é extremamente fundamental que o mapeamento gerado não seja "poluído", ou melhor, não tenha uma quantidade exagerada de informações que possa inviabilizar a leitura e a interpretação dos dados.

O MapServer é uma ferramenta de fácil uso e bastante difundida entre os usuários de SGBD espaciais. Porém, não oferece suporte à três tipos de dados:

- (1, X003, 2) - Polígono simples composto por vértices conectados por arcos circulares;

- (1, 4, x) - Linha composta por alguns vértices conectados por segmentos de reta e outros por arcos;
- (1, x005, x) - Polígono composto por alguns vértices conectados por segmentos de reta e outros por arcos.

No entanto, essas deficiências já estão sendo estudadas pela comunidade MapServer e nas próximas versões possivelmente serão disponibilizadas.

Para a otimização de consultas, o SGBD Oracle permite dois tipos de indexação: R-tree e Quad-tree. É importante ressaltar que quando a base de dados sofre muitas alterações o índice é danificado, sendo necessário uma reconstrução, o que é permitido no SGBD Oracle.

REFERÊNCIAS

CGI. Disponível em: <<http://ms.gis.umn.edu/docs/reference/cgi/referencemanual-all-pages>>. Acesso em: 15/Jul./2005

CIFERRI, R. R. **Análise da Influência do Fator Distribuição Espacial dos Dados no Desempenho de Métodos de Acesso Multidimensionais**. 2002. (Tese de Doutorado). Universidade Federal de Pernambuco. Recife. 2002.

CYRAN, M. **Oracle Database Concepts 10g Release**. Oracle Corporation, dec. 2003.

GATTI, S. D. **Fatores que Afetam o Desempenho de Métodos de Junções Espaciais: um estudo baseado em dados reais**. 2000. (Dissertação de mestrado). UNICAMP. Campinas. 2000.

GUTMAN, A. R-Trees - A dynamic index structure for spatial searching. **ACM Press**, New York, p. 47-57, 1984.

HOEL, E. G.; SAMET, H. A qualitative comparison of data structure for large line segment databases. **ACM Sigmod Conference**. San Diego, p. 205-214, 1992.

MapServer New Users. Disponível em: <http://ms.gis.umn.edu/new_users>. Acesso em: 20/Jul./2005

MURRAY, C. **Oracle Spatial User's Guide and Reference**, Release 9.2. Oracle Corporation, Mar 2002.

OGR Simple Feature Library. Disponível em: <<http://www.gdal.org/ogr/>>. Acesso em: 24/Jul./2005

Oracle Spatial. Disponível em: <http://www.gdal.org/ogr/drv_oci.html>. Acesso em: 25/Jul./2005

RIGAUX, P.; SCHOLL, M.; VOISARD, A. **Spatial Databases with application to GIS**. San Francisco: Morgan Kaufmann , 2002. p. 408

SILVA, R. **Banco de Dados Geográficos: uma análise das arquiteturas dual (Spring) e integrada (Oracle Spatial)**. 2002. (Dissertação de mestrado). USP. São Paulo. 2002.

SIMON, F. **Grupo de discussão MapServer**. Disponível em: <http://br.groups.yahoo.com/group/mapserver_brasil/message/1729>. Acesso em: 23/Jul./2005

Using Oracle Spatial data with MapServer. Disponível em: <http://ms.gis.umn.edu/docs/howto/oracle_spatial_howto>. Acesso em: 10/Jul./2005

VOLKER, G.; OLIVER, G. Multidimensional access methods. **ACM Press**. New York, v.30, no. 2, p. 170-231, 1998.