



UNIVERSIDADE ESTADUAL DA PARAÍBA
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
CENTRO DE CIÊNCIAS E TECNOLOGIA
ESPECIALIZAÇÃO EM GEOPROCESSAMENTO

MARCELO MARCOS RODRIGUES LEITE

**UM ESTUDO COMPARATIVO DOS SISTEMAS DE BANCO DE
DADOS ORACLE E POSTGRESQL COM ÊNFASE NAS
EXTENSÕES ESPACIAIS**

CAMPINA GRANDE - PB

2011

MARCELO MARCOS RODRIGUES LEITE

**UM ESTUDO COMPARATIVO DOS SISTEMAS DE BANCO DE
DADOS ORACLE E POSTGRESQL COM ÊNFASE NAS
EXTENSÕES ESPACIAIS**

Monografia apresentada à Banca Examinadora como requisito para a obtenção do título de Especialista, pelo Programa de Especialização em Geoprocessamento.

Área de Concentração: Geoprocessamento

Orientador: Prof. Dr. Cláudio de Souza Baptista

CAMPINA GRANDE - PB

2011

L533e

Leite, Marcelo Marcos Rodrigues.

Um estudo comparativo dos sistemas de banco de dados Oracle e Postgresql com ênfase nas extensões. [manuscrito] / Marcelo Marcos Rodrigues Leite. - 2011.

80 f. : il. color.

Monografia (Especialização em Geoprocessamento) – Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2011.

“Orientação: Prof. Dr. Cláudio de Souza Baptista, Departamento de Computação”.

1. Sistema gerenciador de banco de dados. 2. Sistema de informação geográfica. 3. PostGIS. I. Título.

21. ed. CDD 005.13

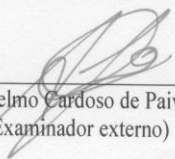
MARCELO MARCOS RODRIGUES LEITE

**UM ESTUDO COMPARATIVO DOS SISTEMAS DE BANCO DE DADOS
ORACLE E POSTGRESQL COM ÊNFASE NAS EXTENSÕES ESPACIAIS**

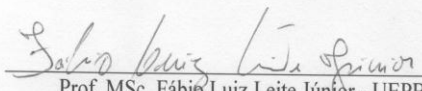
Monografia apresentada à Banca Examinadora como
requisito para a obtenção do título de Especialista,
pelo Programa de Especialização em
Geoprocessamento
Área de Concentração: Geoprocessamento

Aprovado em 9 de novembro de 2011.

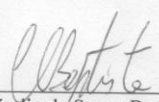
Banca Examinadora



Prof. Dr. Anselmo Cardoso de Paiva - UFMA
(Examinador externo)



Prof. MSc. Fábio Luiz Leite Júnior - UEPB
(Examinador interno)



Prof. Cláudio de Souza Baptista, Ph.D. - UFCG
(Orientador)

CAMPINA GRANDE-PB

2011

Dedico este trabalho à minha esposa Silvia Leite pelo carinho, compreensão e por estar sempre ao meu lado, aos meus filhos Marcelo Leite Filho e Daniel Leite pelo companheirismo e amizade, aos meus pais Eufrosino Leite e Hagar Rodrigues Leite pelo esforço em me educar.

AGRADECIMENTOS

A Deus pela minha vida, por ter me guiado pelo caminho correto e pelas experiências que me proporcionou até hoje.

A minha esposa pela grande ajuda, dedicação, incentivo, estímulo e compreensão. Pelos sacrifícios realizados em prol da conclusão deste trabalho.

Aos meus filhos que estiveram ao meu lado nas horas em que precisei de ajuda ao longo da elaboração deste trabalho.

Aos meus Pais (em memória) pelo carinho, pelas horas de dedicação, pelo incentivo que sempre recebi deles.

Ao orientador, Prof. Dr. Cláudio Baptista, pelo incentivo e apoio, direcionando minhas ações na construção do meu trabalho.

A todos os professores do curso de Geoprocessamento que com seus ensinamentos nos proporcionaram a abertura de novos horizontes.

A todos os amigos da turma do curso de Geoprocessamento pela boa convivência durante os dez meses de curso.

A Vinícius Queiroga e a todos da coordenação do curso, pela sabedoria demonstrada na boa condução deste curso que forma a primeira turma.

Ao Chefe Rogério Fernandes, pela paciência e compreensão em sua longa espera pela minha volta às atividades à plena carga.

E a CHESF, pela oportunidade de me proporcionar um curso de pós-graduação deixando uma marca significativa na minha formação profissional.

“A Ciência nos diz quem somos. A Sabedoria
nos diz quem poderemos ser”

ISAAC NEWTON

RESUMO

Cada vez mais, em ambientes computacionais de empresas e em diversos domínios do conhecimento humano, tem-se observado uma crescente utilização de dados espaciais em sistemas de geoinformações para subsidiar análises e tomadas de decisões. Essa tendência provocou a evolução dos sistemas de gerenciamento de bancos de dados (SGBD) que passaram a ter a característica de extensibilidade com a incorporação de novos tipos de dados não convencionais, dentre eles os dados espaciais. Em Sistemas de Informações Geográficas (SIG), a extensibilidade significa a possibilidade do desenvolvimento de uma extensão geográfica capaz de armazenar, recuperar e analisar dados espaciais. Essa nova característica dos SGBDs permitiu a ampliação da utilização dos SIGs. Atualmente, já existem diversas extensões de SGBDs convencionais que suportam dados não convencionais. Este trabalho apresenta um comparativo entre duas extensões espaciais de SGBDs convencionais, uma da comunidade de software aberto (*PostGIS*) e outra comercial (*Oracle Spatial*). O estudo consiste em analisar as funcionalidades espaciais de cada uma das extensões sob a ótica das especificações *Simple Features Specification for SQL (SFSSQL)* e o padrão internacional *ISO / IEC 13249-3 SQL / MM Part 3: Spatial*.

Palavras-chave: SGBD Objeto-Relacional, Oracle Spatial, PostGIS, Comparação

ABSTRACT

Increasingly, in companies' computer environments and in different fields of human knowledge there has been an increasing use of spatial data in geoinformation systems to provide analysis and decision making. This trend led to the development of Database Management Systems (DBMS) that now have the feature of extensibility to incorporate new types of unconventional data, including spatial data. In Geographic Information Systems (GIS), extensibility means the possibility of developing a geographic area capable of storing, retrieving and analyzing spatial data. This new feature of the DBMS allowed to extend the use of GIS. Nowadays, there are several extensions of conventional DBMS that support non-conventional data. This work presents a comparison between two spatial extensions of conventional DBMS, one of the community of open source software (*PostGIS*) and another commercial (*Oracle Spatial*). The study is to analyze the spatial features of each of the extensions from the perspective of the specifications *Simple Features Specification for SQL (SFSQL)* and the international standard *ISO / IEC 13249-3 SQL / MM Part 3: Spatial*.

Key-Words: Object-Relational DBMS, Oracle Spatial, PostGis, Comparison

LISTA DE FIGURAS

Figura 1 Arquitetura de um sistema de banco de dados simplificado.....	18
Figura 2 Comparação entre SGBD e Sistema de Arquivo Convencional.....	18
Figura 3 Arquitetura em Camadas.....	23
Figura 4 Arquitetura Dual.....	24
Figura 5 Arquitetura Integrada.....	26
Figura 6 Matriz de 9-Interseções para relações entre duas regiões.....	29
Figura 7 Relações Topológicas entre Diferentes Geometrias.....	29
Figura 8 Processamento de consultas espaciais.....	31
Figura 9 Comando <i>SELECT</i> em uma tabela relacional.....	33
Figura 10 Hierarquia de tipos de geometrias da SFSSQL.....	35
Figura 11 Hierarquia de tipos do SQL/MM Spatial.....	37
Figura 12 Tipos espaciais primitivos do Oracle Spatial.....	40
Figura 13 Tipos de dados espaciais do PostGIS.....	44

LISTA DE TABELAS

Tabela 1 Comandos em SQL.....	32
Tabela 2 Quadro Comparativo das extensões <i>Oracle Spatial</i> versus <i>PostGis</i>	49

LISTA DE ABREVIATURAS

BD	Banco de Dados
BLOB	Binary Large Object
CHESF	Companhia Hidroelétrica do São Francisco
DDL	Data Definition Language
DML	Data Manipulation Language
GML	Geographic Markup Language
LAT	Latitude
LONG	Longitude
MBR	Minimum Bounding Rectangle
OGC	Open Geospatial Consortium
PL/SQL	Procedural Language/Structured Query Language
SFSQL	Simple Feature Specification for SQL
SGBD	Sistema Gerenciador de Banco de Dados
SGBD-OR	Sistema Gerenciador de Banco de Dados Objeto Relacional
SGBD-R	Sistema Gerenciador de Banco de Dados Relacional
SGBDE	Sistema Gerenciador de Banco de Dados Espacial
SHP	Shape
SIG	Sistema de Informação Geográfica
SQL	Strutured Query Language
SQL/MM	Strutured Query Language/Multmidia
SRID	Spatial Reference System Identifiers
WKB	Well-Known Binary
WKT	Well-Known Text

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.1.1	Objetivo Geral.....	15
1.1.2	Objetivos Específicos	15
2	BANCO DE DADOS	16
2.1	Conceitos de Bancos de Dados.....	16
2.2	Conceitos de Bancos de Dados Relacionais	19
2.3	Conceitos de Bancos de Dados Objeto Relacional	20
2.4	Banco de Dados Espaciais	22
2.4.1	Arquitetura de Banco de Dados Espaciais.....	23
2.4.1.1	Arquitetura em Camadas	23
2.4.2	Representação de Dados Espaciais	26
2.4.2.1	Representação Vetorial.....	26
2.4.2.2	Representação Matricial (Raster).....	27
2.4.3	Operações Espaciais	27
2.4.4	Métodos de Indexação Espacial	30
2.4.5	Padronização em Banco de Dados	31
2.4.5.1	Padrão SQL.....	31
2.4.5.2	Padrão SFSSQL	34
2.4.5.3	Padrão SQL/MM Spatial	36
2.4.6	Extensões Espaciais de SGBD.....	39
2.4.6.1	Oracle Spatial.....	39
2.4.6.2	PostGreSQL/PostGis	43
3	ESTUDO COMPARATIVO	47
3.1	Comparação entre as Extensões Espaciais <i>Oracle Spatial</i> e <i>PostGIS</i>	47
3.2	Cenários para Consultas Espaciais	50
3.2.1	Resultados e Análises	63
4	CONCLUSÃO	67
5	REFERÊNCIAS BIBLIOGRÁFICAS	68
	ANEXO A	70
	ANEXO B	79

1 INTRODUÇÃO

O processamento e a análise de dados espaciais estão cada vez mais dependentes do uso de sistemas de gestão de banco de dados (SGBD) ao invés de SIG convencionais. Esta abordagem para o tratamento de dados espaciais facilita a relação custo-eficácia no gerenciamento de dados usando os recursos de SGBDs. Dentre estes recursos podem ser destacados armazenamento e indexação de informação, sistemas de segurança, tolerância a falhas, integridade de dados, controle de concorrência, backup e recuperação, controle de distribuição e replicação de dados e, mais importante, uma estreita integração com ambientes de computação empresarial corporativo. Com isso as empresas estão passando a usar o processamento de dados espaciais mais como um componente de tecnologia da informação (TI) corporativa do que como tradicionais aplicativos departamentais *stand alone*.

O uso de SGBD não só mudou os conceitos fundamentais de dados espaciais e seu gerenciamento, mas também criou uma demanda por novas competências e uma nova geração de gestores e usuários de dados espaciais, nos quais se espera a competência nos princípios e métodos dessa relativamente complexa tecnologia de banco de dados. Os sistemas de banco de dados espaciais são diferentes dos convencionais sistemas de banco de dados em dois aspectos importantes. O primeiro é a necessidade de armazenar tipos de dados complexos, como pontos, linhas e polígonos. O segundo é a funcionalidade necessária para processar esses tipos de dados complexos, utilizando os operadores espaciais que são consideravelmente mais sofisticadas do que os operadores de banco de dados convencionais para o tratamento dos tipos de dados alfanuméricos. As linguagens de consulta *SQL(Structured Query Language)* são padrões especificados cujas implementações em SGBD relacionais buscam cumprir esse papel de armazenamento e manipulação dos dados. Os SGBD espaciais estendem *SQL* para transformá-la em linguagem para consulta espacial.

Atualmente, o mercado trabalha principalmente com duas tecnologias de SGBDs chamadas SGBD Relacional (SGBD-R) e SGBDs Objeto-Relacionais (SGBD-OR). Ainda há uma terceira tecnologia de SGBD chamada SGBDs Orientados-a-Objeto (SGBD-OO) que fica com uma pequena parcela do mercado (QUEIROZ et al., 2006).

Os SGBD-R foram concebidos para atender às necessidades de aplicações manipulando grandes volumes de dados convencionais. De fato, tais sistemas não oferecem recursos para atender as necessidades de aplicações não convencionais. A mera simulação de tipos de dados não convencionais em um SGBD-R pode ter efeitos colaterais, como queda de

desempenho, dificuldade de codificação e posterior manutenção da aplicação (QUEIROZ et al., 2006 apud STONEBRAKER, 1996).

No que se refere à representação de dados espaciais, são agrupados em duas grandes classes ou modelos de representação, a saber: vetorial e matricial (*raster*). Este trabalho trata apenas da representação de dados vetorial.

Este trabalho volta a atenção para sistemas de banco de dados espaciais apresentando um estudo comparativo das arquiteturas *Oracle Spatial* e *PostGresSQL/PostGis*, que são extensões espaciais desenvolvidas sobre Sistemas de Banco de Dados Objeto-Relacional (SGBD-OR). Assim, serão analisados principalmente os tipos de dados espaciais, novas funções e operadores incorporados à linguagem SQL para manipulação desses dados espaciais à luz das normas *Simple Features Specification For SQL (SFSSQL)* e *ISO/IEC 13249-3 SQL Multimídia and Application Packages (SQL/MM) – Part 3: Spatial* e técnicas de indexação espacial.

Como parte ilustrativa desse trabalho, foram criados cenários para demonstrar a utilização da linguagem de consulta *SQL* com alguns operadores espaciais. Para a realização das consultas aos dados espaciais foi criado um banco de dados de teste a partir do banco de dados do *SmartView*. O *SmartView* é um sistema de visualização de informações, em um ambiente geográfico, que fornece suporte à Operação do Sistema Elétrico da CHESF. As consultas aos dados espaciais foram projetadas e executadas, tanto em *Oracle Spatial* como em *PostGresSQL/PostGIS*. Os consultas para os referidos cenários se encontram no ANEXO A. No ANEXO B encontra-se uma descrição do sistema *SmartView* da CHESF e do seu banco de dados.

Como este trabalho visa analisar e apresentar as principais características de SGBD-OR com suporte a dados espaciais, e não desenvolver análises reais sobre a rede elétrica do sistema elétrico da CHESF, não se faz necessário que a base de dados seja real. Portanto, o banco de dados de teste usado para as consultas pode conter alguns dados, não reais, inseridos manualmente.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é apresentar um estudo comparativo envolvendo os Sistemas de Banco de dados objeto-relacionais com suas extensões espaciais: *Oracle Spatial* e *PostGreSQL/PostGis* à luz das normas *SFSSQL* e *SQL/MM Spatial*.

1.1.2 Objetivos Específicos

1. Disponibilizar informações sobre as extensões do padrão de linguagem de acesso *SQL* para fornecer suporte à manipulação e consulta de dados em SGBDs com extensões espaciais.
2. Prover informações sobre as características e modelos de dados espaciais implementados para os SGBDs espaciais comparados.
3. Fornecer subsídios que possibilitem aos gerentes de tecnologia de informação bem como aos desenvolvedores de sistemas de informação geográfica, interessados em implantar bancos de dados geográficos na CHESF e em outras empresas do setor elétrico, identificar aquele SGBD que pode ser mais bem utilizado no desenvolvimento de aplicações de geoprocessamento.

Os capítulos seguintes deste trabalho estão organizados da seguinte forma:

Capítulo 2: revisa os conceitos de Banco de Dados em geral; apresenta Arquitetura de Banco de Dados espaciais; aborda conceitos de Representação de dados espaciais; descreve Métodos de indexação espacial; apresenta Padronização em banco de dados espaciais e as subseções 2.4.6.1 e 2.4.6.2 abordam conceitos das extensões espaciais *Oracle Spatial* e o *PostGis* respectivamente;

Capítulo 3: apresenta um estudo comparativo entre *Oracle Spatial* e *PostGis* mostrando suas características e funcionalidades espaciais tendo como balizadores as normas *SFSSQL* e *SQL/MM Spatial*;

Capítulo 4: apresenta algumas conclusões e um possível trabalho a ser realizado para o enriquecimento do atual.

2 BANCO DE DADOS

Neste capítulo é feita uma abordagem sobre os conceitos e modelos de Sistemas de Gerenciamento de Banco de Dados (SGBD), a representação de dados geográficos, os relacionamentos topológicos, as funções espaciais, as arquiteturas de Banco de Dados Espaciais, os métodos de acesso, os padrões de SQL e as extensões espaciais Oracle Spatial e PostGis.

2.1 Conceitos de Bancos de Dados

Banco de dados é uma coleção de dados relacionados e armazenados em algum dispositivo. As propriedades de um banco de dados são: coleção de dados com algum significado inerente; é projetado, construído e “povoado” com dados para um específico propósito e representa um mini-mundo (BAPTISTA, 2010).

Os dados são fatos que podem ser gravados e que possuem um significado implícito. Por exemplo, nomes, números telefônicos e endereços de pessoas. Esses dados podem ter sido escritos em uma agenda de telefones ou armazenados em um computador. Essas informações são coleções de dados com um significado implícito caracterizando um banco de dados.

A organização de um banco de dados envolve, basicamente, um mecanismo eficiente de armazenamento e manipulação de dados. Para facilitar a administração surgiram os Sistemas de Gerenciamento de Banco de Dados (SGBD). Um SGBD, segundo BAPTISTA (2010) e QUEIROZ (et al., 2006), é um *software* que permite construir e manipular um banco de dados oferecendo suporte para o armazenamento, consulta e atualização de bancos de dados. Ainda segundo QUEIROZ (et al., 2006) atualmente o mercado concentra-se em duas tecnologias de SGBD, que são os relacionais (SGBD-R) e os objeto-relacionais (SGBD-OR).

Segundo Date (1990) o sistema gerenciador de banco de dados (SGBD) é o *software* que manipula todos os acessos ao banco de dados. O usuário emite uma solicitação de acesso, usando uma sublinguagem específica de dados (por exemplo, SQL). Ainda para Baptista (2010) um SGBD é o *software* que permite construir e manipular um banco de dados. São exemplos de SGBD os seguintes: Oracle, SQL Server, DB2, PostGreSQL, MySQL e Sybase.

Conforme DATE (1990) pode-se caracterizar que a função do SGBD é proporcionar a interface de usuário ao sistema de banco de dados. A interface de usuário pode ser definida como limite no sistema abaixo do qual tudo é invisível.

Os SGBD são sistemas complexos e possuem cada vez mais funcionalidades. As principais vantagens conforme BAPTISTA (2010) são:

- Controlar redundância
- Compartilhamento de dados
- Independência de dados
- Segurança
- Backup e recuperação a falhas
- Forçar restrições de integridade
- Aumentar a produtividade e a disponibilidade
- Flexibilidade, padronização.

A Figura 1 mostra uma arquitetura típica de um Sistema Gerenciador de Banco de Dados. Na parte inferior, os elementos cilíndricos representam o banco de dados estático, tal como ele é fisicamente armazenado em disco. Observa-se que juntamente com os dados estão os metadados, isto é, um conjunto de dados que contém informação sobre a estrutura dos dados, também chamado de dicionário de dados ou catálogo do sistema. O software para acesso aos dados armazenados (Gerenciador de Armazenamento) faz a interface entre os dados armazenados e o programa para processamento de consultas (Processador de Consultas), capturando do disco o(s) bloco(s) de dados necessários para atender aos requisitos de uma consulta. O Processador de Consultas interpreta a Linguagem de Manipulação de Dados (DML) utilizada pelo banco de dados, gerando um plano estratégico de execução para que o Gerenciador de Armazenamento possa efetuar as operações necessárias. A parte da DML responsável pela recuperação de informações é chamada Linguagem de Consulta SQL (*Structured Query Language*). Apesar do nome, o Processador de Consultas é responsável tanto por processar as consultas como as requisições de modificação nos dados e nos metadados. O SGBD deve definir um plano estratégico de execução para recuperar o resultado da consulta e passar o controle e as devidas instruções ao Gerenciador de Armazenamento. O SGBD que possui recursos para garantir a integridade do sistema assegura que várias consultas sendo executadas simultaneamente não interfiram umas com as outras e que não haverá perda de dados mesmo em caso de falhas.

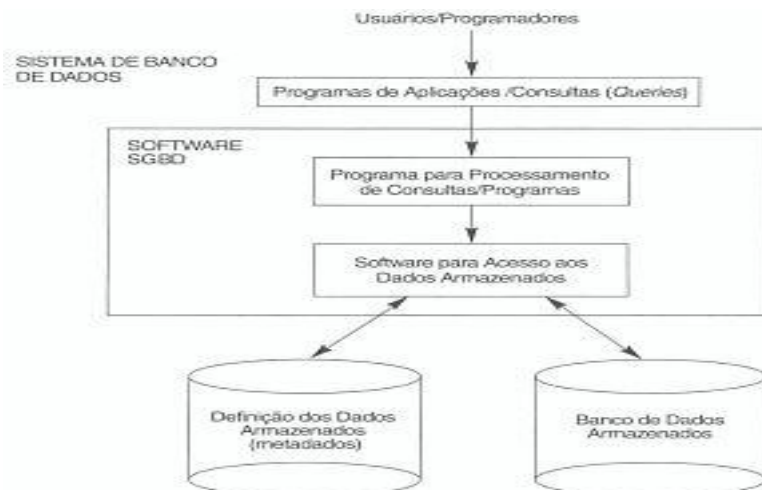


Figura 1: Arquitetura de um sistema de banco de dados simplificado
Fonte: ELMASRI, NAVATHE. Sistemas de banco de dados. 2005.

BAPTISTA (2010) apresenta um comparativo entre um SGBD e um Sistema de Arquivo conforme mostrado na figura 2.



Figura 2: Comparação entre SGBD e Sistema de Arquivo Convencional
Fonte: BAPTISTA (2010).

Um modelo de dados é um conjunto de conceitos para descrever um BD. De acordo com BAPTISTA(2004), existem três categorias de modelo de dados: modelo conceitual, modelo lógico e modelo interno.

Um modelo conceitual é aquele que não é utilizado pelos SGBD, mas é importante na chamada fase conceitual do projeto de um BD. Para a elaboração da modelagem semântica das estruturas de dados mantidas em um banco de dados, destaca-se

dentre outros o MER (Modelo de Entidades e Relacionamentos) (CHEN,1976) e suas variações.

Modelos lógicos são modelos de dados utilizados pelo sistema de gerência de banco de dados (SGBD). São lógicos porque sua implementação não precisa ser conhecida. Exemplos: Modelo Relacional, Modelos Orientados a Objeto Jasmine, GemStone, etc.

Um modelo interno (também chamado de modelo físico) é aquele que é usado para implementar um modelo lógico. Cada SGBD tem o seu modelo interno.

Os modelo físicos de dados descrevem os detalhes de como os dados estão armazenados no computador, representando informações como formato de registros, ordenamento de registros e caminhos de acesso. Neste modelo existe a linguagem SQL(Structured Query Language) para consultas em bancos de dados.

O modelo de dados lógico Relacional é atualmente o mais utilizado nos SGBDs comerciais. Entretanto, este modelo possui um sistema de tipos simples e restrito, o que dificulta a descrição de algumas aplicações atuais que necessitam tipos mais complexos e características do modelo Orientado a Objetos.

O modelo Objeto Relacional surgiu a partir destas novas necessidades das aplicações atuais. Desta forma ele estende o modelo Relacional adicionando algumas destas características.

Sistemas de Banco de Dados que utilizam o modelo relacional, ou seja, SGBD-R são também considerados sistemas da segunda geração de SGBD visto que os sistemas de Banco de Dados Hierárquicos e de Rede são considerados a primeira geração. Assim, os sistemas de Banco de Dados Objeto Relacionais são classificados como a terceira geração de SGBD.

Adiante será mostrado que o modelo Objeto Relacional é baseado no modelo Relacional e no modelo Orientado a Objeto.

2.2 Conceitos de Bancos de Dados Relacionais

Com a evolução dos bancos de dados, vários modelos de implementação têm sido propostos onde se destaca o modelo relacional. Um banco de dados relacional é formado por uma coleção de tabelas que admite operações sobre elas. Dentre essas operações, podem-se citar as de atualização, inserção, exclusão e seleção. As tabelas (relações) armazenam os registros (tuplas) e implementam os relacionamentos entre os dados.

O modelo relacional tornou-se um padrão de fato para aplicações comerciais, devido a sua simplicidade e performance. É um modelo formal, baseado na teoria matemática das relações. O modelo relacional representa os dados num BD como uma coleção de tabelas (relações); Cada tabela terá um nome, que será único, e um conjunto de atributos com seus respectivos nomes e domínios; Todos os valores de uma coluna são do mesmo tipo de dados (domínio) que deve ser atômico, ou seja, seus valores devem ser indivisíveis. Na terminologia de BD Relacional, uma linha é chamada tupla, um nome de coluna é chamado de atributo e cada tabela é chamada de relação. (BAPTISTA, 2010).

O modelo relacional foi originalmente descrito por E. F. Codd, um pesquisador dos laboratórios da IBM, no final da década de 60. A partir disto vários fabricantes de software como a IBM, por exemplo, implementaram esse modelo em seus produtos. O modelo relacional foi o primeiro modelo de dados utilizado em aplicações comerciais, e continua sendo utilizado na maioria das aplicações atuais.

Em um banco de dados relacional além das relações, compõem também sua especificação as chaves e restrições de integridade. Chaves servem para identificar tuplas e definir o relacionamento entre tuplas de relações. Podem ser do tipo chave primária, chave estrangeira ou ainda chave alternativa. As Restrições de Integridade são regras de consistência de dados. A descrição completa do modelo relacional pode ser encontrada em CODD (1999).

Nos sistemas comerciais atuais, os tipos incluem números inteiros, de ponto flutuante, cadeias de caracteres, datas e campos binários longos (BLOBs). Para esses tipos encontram-se disponíveis uma variedade de operações (exceto para o tipo BLOB), como operações aritméticas, de conversão, de manipulação textual e operações com data. (FERREIRA, et al., 2005). Os Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBD-R) foram desenvolvidos para atender as necessidades de sistemas que manipulavam grandes volumes de dados alfanuméricos.

2.3 Conceitos de Bancos de Dados Objeto Relacional

Novas aplicações que surgiram com necessidades específicas não eram atendidas de forma adequada pelos SGBDs relacionais. Aplicações como multimídia e aplicações geográficas, precisam representar seus dados de forma mais real, o que implica em características não existentes no modelo relacional, como tipos de dados mais complexos.

O modelo orientado a objetos baseia-se em extensões do conceito de tipos de dados abstratos, incluindo estruturas mais complexas, disponibilidade de novos tipos de dados e a definição de operações sobre esses dados. Em *ODBMS (Object Database Management Systems* – <http://www.odbms.org>), encontram-se padrões para armazenamento de objetos e o estado da arte sobre Sistemas Gerenciadores de Banco de Dados Orientados a Objeto (SGBD-OO).

Os SGBDs Objeto-Relacionais (SGBD-OR) são sistemas que surgiram como uma alternativa para acomodar a evolução das aplicações bem como para atender as suas novas necessidades. Este tipo de SGBD está em uma categoria entre os SGBD-R e os SGBD-OO, pois ao mesmo tempo em que permite a utilização de estruturas de dados complexas, não deixa de priorizar o acesso declaratório aos dados. Os dados continuam sendo armazenados na forma de relações com tuplas e atributos, porém o valor de um atributo em determinada tupla pode ser de um tipo de dado definido pelo usuário, por exemplo, ou uma outra relação. Esta característica estende o poder da modelagem.

Os SGBDs Objeto-Relacionais (SGBD-OR) permitem a integração de dados convencionais com outros tipos de dados. O modelo puramente Relacional só permitia tipos de dados simples, tais como: inteiros, reais, *strings*, datas, entre outros. A abordagem objeto-relacional, da mesma forma que os SGBDs Orientados a Objeto, suporta objetos complexos, além dos simples. Com os tipos de dados simples mais os BLOBs da abordagem relacional e através dos conceitos para adição de novos tipos de dados (Tipos Abstratos de Dados – TDA) e da orientação a objetos, é possível, aos SGBD-ORs, o suporte a objetos complexos, como é o caso dos dados espaciais e multimídia. Possibilitam ainda a extensão dos mecanismos de indexação sobre os novos tipos, como por exemplo os TDEs (Tipos de Dados Espaciais).

No contexto deste trabalho, é dada ênfase aos SGBD-ORs. As extensões espaciais desenvolvidas sobre SGBD-OR permitem definir tipos de dados espaciais com operadores e métodos de acesso específicos como as extensões que serão estudadas neste trabalho, *Oracle Spatial* e o *PostGIS*.

No escopo deste trabalho interessa verificar maiores detalhes do padrão *SQL* enquanto linguagem (partes *DDL* e *DML*) para definição e manipulação de dados em SGBD-OR aplicados à Banco de dados Espaciais. Um padrão revisado e mais expandido chamado *SQL2* ou *SQL-92* foi desenvolvido para consultas em bancos de dados relacionais, enquanto a *SQL3* estende ainda mais a *SQL* com SGBD-OR. (ELMASRI e NAVATHE, 2004).

Assim em subseções posteriores serão apresentadas algumas características da

linguagem *SQL* e serão apresentados os padrões para banco de dados espaciais *SFSSQL* e *SQL/MM SPATIAL*.

Os SGBDs Oracle e PostGreSQL são dois exemplos de bancos de dados objeto-relacionais. (FERREIRA, et al., 2005). No caso de dados espaciais, existem extensões para modelagem física em ambos os SGBD-ORs, baseadas nas especificações do OGC, porém existem variações relevantes entre os modelos internos de dados, semântica dos operadores espaciais, mecanismos de indexação e esquema de sintaxe da *SQL* estendida com tipos espaciais.

2.4 Banco de Dados Espaciais

Sistemas de Gerência de Banco de Dados Espaciais (SGBDE) permitem o gerenciamento e processamento de dados referenciados geograficamente. Bancos de Dados Objeto Relacional tem se tornado o novo padrão para tratar o crescente gerenciamento de dados e análise das necessidades de aplicações de banco de dados não-tradicionais, como os Sistemas de Informações Espaciais. Duas questões críticas devem ser resolvidas a fim de satisfazer eficazmente as necessidades dessas aplicações. São elas: a representação e pesquisa baseada em conteúdo de dados espaciais. Os SGBDs com extensões espaciais resolvem esses problemas, provendo um tipo de dado espacial, um mecanismo de indexação, e funções/operadores para os tipos de dados espaciais.

Para VINHAS (2006), Bancos de dados geográficos são coleções de dados georreferenciados, manipulados por Sistemas de Informação Geográfica (SIG). Os SIGs são sistemas computacionais capazes de capturar, modelar, armazenar, recuperar, manipular, analisar e apresentar dados geográficos.

Dentre as finalidades e possibilidades que as bases de dados com geometria oferecem podemos citar as de análise e consultas espaciais. É possível calcular, por exemplo, áreas, distâncias e centróides, além de realizar a geração de buffers e outras operações entre as geometrias. Porém quando se trata da sua manipulação e representação envolvem tarefas bastante complexas.

Nas seções seguintes serão abordados aspectos sobre Arquiteturas de Banco de Dados Espaciais, Representação de Dados Espaciais, Funções Espaciais, Indexação Espacial e Extensões Espaciais.

2.4.1 Arquitetura de Banco de Dados Espaciais

A existência de diferentes arquiteturas de banco de dados geográficos se deve à necessidade de integração entre dados convencionais e dados espaciais. Essa integração é importante, senão fundamental, porque permite a análise conjunta de vários tipos de informações e onde elas ocorrem no espaço (SILVA, 2002).

A integração de dados espaciais com dados convencionais foi uma preocupação posterior ao desenvolvimento das ferramentas SIG e dos SGBDs convencionais. A necessidade de integração impulsionou o desenvolvimento de novos mecanismos computacionais de conexão entre esses dois tipos de dados (SILVA, 2002).

A seguir, essas abordagens e suas implicações são apresentadas em mais detalhes.

2.4.1.1 Arquitetura em Camadas

BAPTISTA (2010) apresenta uma abordagem sobre arquitetura de um SIG comumente empregada para descrever sistemas computacionais que consiste em separá-los em camadas funcionalmente cada vez mais complexas. Descreve sucintamente uma arquitetura que separa as camadas de: Armazenamento, Manipulação e Visualização.

A Figura 3 mostra as camadas e subsistemas dessa arquitetura proposta.

SV		SM	
SMM	SMV	SMC	
SAM	SAV	SAC	
SAF			

SV Sub sistema de Visualização

SMM Sub sistema de Manipulação Matricial

SAM Sub sistema de Armazenamento Matricial

SAF Sub sistema de Armazenamento Físico

SM Sub sistema de Manipulação

SMV Sub sistema de Manipulação Vetorial

SAV Sub sistema de Armazenamento Vetorial

SMC Sub sistema de Manipulação Convencional

SAC Sub sistema de Armazenamento Convencional

Figura 3 Arquitetura em Camadas

Fonte: BATISTA, 2010

A primeira camada trata da Visualização e Manipulação:

- A Visualização oferece funções básicas para visualização de objetos tradicionais e georeferenciados;
- A Manipulação oferece funções para denificação e manipulação destes objetos.

A segunda camada corresponde ao nível de representação do modelo:

- Oferece separadamente serviços de manipulação de alto nível para atributos convencionais;
- Representação raster;
- Representação vetorial.

A terceira camada engloba os subsistemas que oferecem serviços de armazenamento:

- Armazenamento de dados não-espaciais
- Armazenamento de dados em formato raster
- Armazenamento de dados em formato vector
- Fazem uso do subsistema de Armazenamento Físico (SAF) que está na quarta camada.

Ainda segundo BAPTISTA (2010), existem diferentes estratégias de implementação para a arquitetura em camadas, baseadas em sistemas de gerência de bancos de dados com grau crescente de funcionalidade definido como **Relacional, Dual, Campos Longos e Integrada**.

Analisando as estratégias citadas por BAPTISTA (2010), podemos descrever que:

Relacional: Os temas são representados na forma de relações, sendo objeto geográfico uma tupla de uma relação; Atributos são tipos simples; Permite uso de SQL para consulta aos dados; Não há implementação de SV, SAM, SMM. Esta estratégia apresenta algumas desvantagens tais como: dificuldade em elaborar consultas, baixa performance (muitas junções), dificuldade de definir tipos espaciais, impossibilidade de expressar consultas espaciais (SMV e SMM), sendo uma alternativa pouco viável.

Dual: Um SIG usando a arquitetura DUAL ou HÍBRIDA, conforme Figura 4, possui um SGBD relacional para armazenar em tabelas, a componente convencional de todos os objetos (dados não espaciais) e arquivos normais para a componente espacial dos objetos. BAPTISTA (2010).

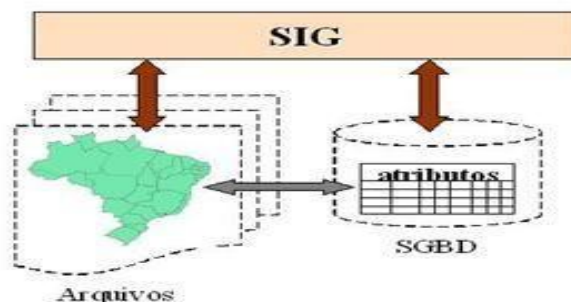


Figura 4 Arquitetura Dual
Fonte: FERREIRA et al., 2005

De acordo com BATISTA (2010), do ponto de vista da arquitetura em camadas:

- SMC e SAC sob um SGBD Relacional;
- SMM, SAM, SMV e SAV sobre o sistema de arquivos do sistema operacional.

Ou seja, separação entre o processamento da parte convencional, realizado pelo SGBD, e o processamento da parte espacial, realizado pelo aplicativo utilizando os arquivos proprietários (FERREIRA, et al., 2005);

BAPTISTA (2010) salienta os problemas da estratégia DUAL, onde a mesma exige treinamento em dois mundos distintos (SGBD + SIG), não utiliza recursos de SGBD (concorrência, tolerância a falhas, otimização, integridade) para os dados espaciais, SAM e SAV só usam as funcionalidades de sistema de arquivo (pobre) e problema de integridade de dados são altos.

Campos Longos: Também chamados de BLOBs, esta estratégia baseia-se no uso de SGBDs relacionais com suporte para campos longos, nos quais são armazenadas as componentes espaciais dos objetos, SMC, SAC, SAV e SAM estão no SGBD, SMM e SMV ainda estão fora do SGBD e é também usado por vários fabricantes.

A vantagem é o uso de um SGBD para representação de dados convencionais e espaciais, fazendo uso de toda funcionalidade de um SGBD. A desvantagem é que um campo BLOB é uma caixa preta (binário) em que se pode acessar todo o objeto ou parte deste, mas sem conhecer a semântica do objeto. Em consequência, a implementação do SIG ainda deve incluir métodos de acesso espacial e um otimizador de consultas que prepare planos de execução que façam uso destes métodos (BAPTISTA (2010)).

Integrada: BAPTISTA (2010) descreve a arquitetura integrada, ilustrado na Figura 5, como sendo o uso de um SGBD extensível (Orientado a Objeto ou Objeto Relacional) que disponha de mecanismos que permitam implementar o tratamento das componentes espaciais através de extensões ao seu ambiente.

BAPTISTA (2010), analisa que, na estratégia integrada, a integração de dados convencionais com espaciais é total, inclusive usando toda funcionalidade provida por um SGBD (integridade, segurança, backup e recuperação dos dados) que na estratégia Dual eram desfrutadas apenas pelos dados descritivos, possuindo semântica de dados espaciais conhecida e os SMM, SAM, SMV e o SAV são extensões do SGBD além de uma melhor performance.



Figura 5 Arquitetura Integrada
Fonte: FERREIRA et al., 2005

São exemplos dessa arquitetura as extensões espaciais apresentadas em (QUEIROZ e FERREIRA, 2005), uma comercial *Oracle Spatial*® (RAVADA & SHARMA, 1999; MURRAY, 2003) e a extensão espacial de licença livre, ou seja, *freeware*, a extensão espacial *PostGIS* (RAMSEY et al., 2010) para o SGBD PostgreSQL, que são analisadas neste trabalho.

2.4.2 Representação de Dados Espaciais

Os **dados geográficos** são diferenciados dos demais pela sua componente espacial. Por isso eles também são chamados de **dados espaciais**. A componente espacial destes dados são representações da superfície terrestre que estão relacionados com sua localização no espaço, ou seja, podem ser posicionados em determinada região geográfica tendo por base suas coordenadas. Sendo assim, torna-se possível a análise do espaço geográfico.

Com relação à representação de dados espaciais, as duas grandes classes de representações geométricas são: Representação Matricial (*raster*) e Representação Vetorial (CÂMARA, 2005).

Estas classes de representação se referem à forma na qual os dados espaciais são armazenados (vetores ou matrizes).

2.4.2.1 Representação Vetorial

As estruturas vetoriais são utilizadas para representar as coordenadas das fronteiras de cada entidade geográfica, através de três formas básicas: pontos, linhas, e áreas (ou polígonos), definidas por suas coordenadas cartesianas (CÂMARA, 2005).

Segundo CÂMARA (2005), um ponto é um par ordenado (x, y) de coordenadas espaciais. Pontos são utilizados para representar, por exemplo, a localização de crimes,

ocorrências de doenças ou torres de transmissão. Uma linha é um conjunto de pontos conectados. Linhas têm aplicação na representação de redes de esgoto, traçado de rios, linhas de transmissão e similares. Uma área (ou polígono) é a região do plano limitada por uma ou mais linhas poligonais conectadas de tal forma que o último ponto de uma linha seja idêntico ao primeiro da próxima. Polígonos podem representar desde lotes de uma quadra até continentes. Com respeito aos polígonos, Observe-se também que a fronteira do polígono divide o plano em duas regiões: o interior, que em geral inclui a fronteira do polígono fechado, e o exterior.

2.4.2.2 Representação Matricial (Raster)

Em CÂMARA (2005), as estruturas matriciais usam uma grade regular sobre a qual se representa, célula a célula, o elemento que está sendo representado. A cada célula, atribui-se um código referente ao atributo estudado, de tal forma que o computador saiba a que elemento ou objeto pertence determinada célula. Nesta representação, o espaço é representado através de uma matriz composta de certo número de colunas e linhas, onde cada célula tem um valor correspondente ao atributo analisado e pode ser individualmente acessada pelas suas coordenadas. A resolução do sistema é dada pela relação entre o tamanho da célula no mapa ou documento e a área por ela coberta no terreno.

A escolha do uso da estrutura vetorial ou matricial dependerá de diversos fatores, pois ambos apresentam vantagens e desvantagens na sua utilização. Por exemplo, a classe *raster* representa melhor fenômenos com variação contínua no espaço. Já o armazenamento na forma de vetores (por coordenadas) é mais preciso.

O armazenamento de dados na representação matricial não será tratado, pois foge do escopo deste trabalho.

2.4.3 Operações Espaciais

Um SGBD Espacial disponibiliza uma variedade de operações prontas para auxiliar na manipulação dos dados espaciais. As operações espaciais são utilizadas pelas Linguagens de Consulta na realização de consultas espaciais.

As consultas espaciais baseiam-se em relacionamentos espaciais. Existem vários tipos de relacionamentos espaciais, mas os mais importantes para um SGBD Espacial são: métricos, direcionais e topológicos. As operações espaciais podem ser classificadas em FERREIRA (et al. 2005 apud RIGAUX et al. 2002):

- a. *Relacionamento topológico*: um relacionamento que não é alterado por transformações topológicas, como translação, rotação e mudança de escala. Como exemplos, temos: contém (contains), disjunto (disjoint), intercepta (intersects), cruza (crosses), etc.
- b. *Relacionamento direcional*: um relacionamento que expressa uma noção de direção. Como exemplos, temos: acima de (above), ao norte de (northOf), dentre outras;
- c. *Relacionamento métrico*: um relacionamento que expressa uma noção métrica. Por exemplo, o relacionamento que retorna Verdadeiro se duas geometrias estão a menos de uma determinada distância uma da outra.

2.4.3.1 Relacionamento Topológico

Ao se trabalhar com dados geográficos, é bastante útil determinar relações topológicas como adjacência, pertinência, intersecção, cruzamento e proximidade. Por exemplo, em determinado projeto pode surgir a questão: Que municípios são cortados pelo Rio São Francisco?

Relacionamento Topológico, em estruturas vetoriais, representam as relações de vizinhança espacial interna e externa dos objetos. Os relacionamentos topológicos são baseados em três abordagens comumente aceitas. Cada uma dessas abordagens descreve o relacionamento entre dois objetos baseada em uma matriz de interseção:

- Matriz de 4-Interseções (4IM);
- Matriz de 9-Interseções (9IM);
- Matriz de 9-Interseções Estendida Dimensionalmente (DE-9IM).

Inicialmente, foram definidas oito relações topológicas binárias baseadas nas quatro interseções entre fronteiras e interiores de dois objetos geográficos n -dimensionais (DAVIS Jr.(b) e QUEIROZ (2005) apud Egenhofer e Franzola, 1995). Este modelo é chamado de 4-Interseções e as interseções entre as fronteiras (∂A) e interiores (A°) dos objetos são analisadas de acordo com seu valor, vazio (\emptyset) ou não-vazio ($\neg\emptyset$).

Para suportar relações espaciais entre objetos geográficos com estruturas mais complexas, como regiões com ilhas e separações, foi necessário acrescentar o conceito de exterior (A^-) de um objeto geográfico ao modelo de 4-Interseções. Portanto, o modelo foi estendido para analisar o resultado da interseção entre as fronteiras, interiores e exteriores de dois objetos, resultando em um novo modelo chamado Modelo de 9-Interseções, modelo ilustrado na Figura 6 (DAVIS Jr.(b) e QUEIROZ, 2005 apud Egenhofer e Herring, 1991).

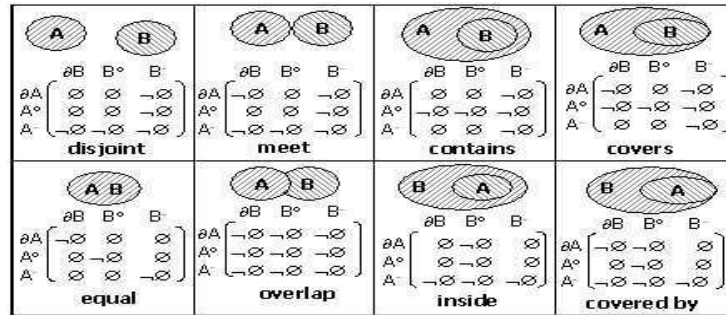


Figura 6: Matriz de 9-Interseções para relações entre duas regiões.
Fonte: DAVIS Jr.(b) e QUEIROZ (2005).

Nos modelos citados acima, ressaltam DAVIS Jr.(b) e QUEIROZ (2005), os resultados das interseções são avaliados considerando os valores vazio ou não-vazio. Há várias situações em que é necessário considerar as dimensões das interseções não vazias.

Com isso, novos modelos foram definidos, que consideram as dimensões dos resultados das interseções não vazias, como o Modelo de 9-Interseções Estendido Dimensionalmente (DE-9IM), que é baseado no modelo de 9-Interseções (DAVIS Jr.(b) e QUEIROZ, 2005). Este modelo contempla assim um conjunto de 52 relacionamentos topológicos, o que não é conveniente do ponto de vista do usuário. Para equacionar este problema, os relacionamentos topológicos foram agrupados em cinco mais gerais – touch, in, cross, overlap, disjoint – que são sobrecarregados, ou seja, que podem ser usados indistintamente para ponto, linha e região. Figura 7.

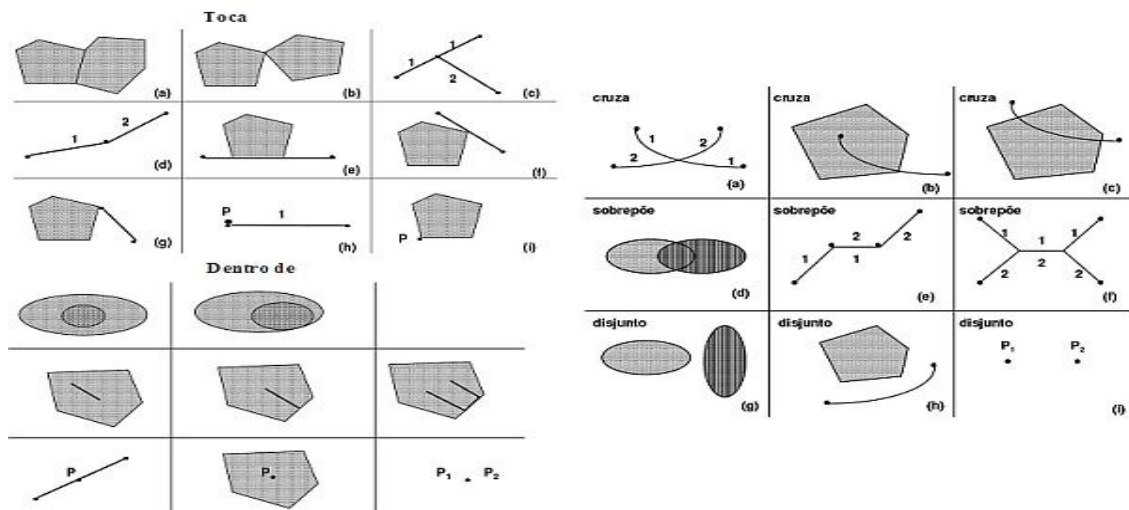


Figura 7: Relações Topológicas entre Diferentes Geometrias
Fonte: <http://wiki.dpi.inpe.br/doku.php?id=cap349/>

2.4.4 Métodos de Indexação Espacial

Bancos de dados, normalmente, estão armazenados em memória secundária cujo acesso é bem mais lento comparado à memória principal. Mecanismos de indexação utilizam os índices que aceleram o processamento das consultas em banco de dados, organizam os dados em árvores de busca que possam rapidamente ser percorridas para encontrar um registro específico.

No caso de bancos de dados convencionais, que tratam dados unidimensionais, os índices aceleram as consultas utilizando estruturas de dados dinâmicas, tais como *Tabelas Hash* e *B-tree* (mais detalhes sobre essas estruturas podem ser encontrados em ELMASRI e NAVATHE (2000)), e possuem como chave de pesquisa um ou vários atributos concatenados. Já no caso dos bancos de dados espaciais, que trabalham com informações de natureza multidimensionais, os índices devem levar em consideração a localização espacial dos objetos.

Os valores das chaves do índice *B-tree* são totalmente ordenados atribuindo ao índice propriedades que permitem eficiência e desta forma esse índice não se adapta facilmente aos dados espaciais. (RIGAUX et al. 2002).

Método de indexação espacial trata-se de técnicas que visam otimizar o acesso aos dados, com o objetivo de acelerar as consultas espaciais. Para tanto, o que normalmente se faz é reduzir o conjunto de dados (espaço de busca) a ser verificado no processamento. Uma idéia fundamental dos índices espaciais é o uso de aproximações, isto é, a estrutura do índice trabalha com representações mais simples dos objetos, como o menor retângulo envolvente (MRE) do objeto. Essas técnicas normalmente empregam a estratégia de processamento de consultas divididas em duas fases: Filtragem e Refinamento (Figura 8).

De acordo com DAVIS Jr.(a) e QUEIROZ (2005), na fase de filtragem são usados métodos de acesso espaciais. O principal objetivo do uso destes métodos é o de reduzir e rapidamente selecionar os possíveis candidatos que satisfaçam a consulta. A redução do espaço de busca é muito importante, pois a fase seguinte, a de refinamento, envolve a aplicação de algoritmos geométricos computacionalmente complexos e custosos e que são aplicados à geometria exata dos candidatos selecionados na etapa anterior.

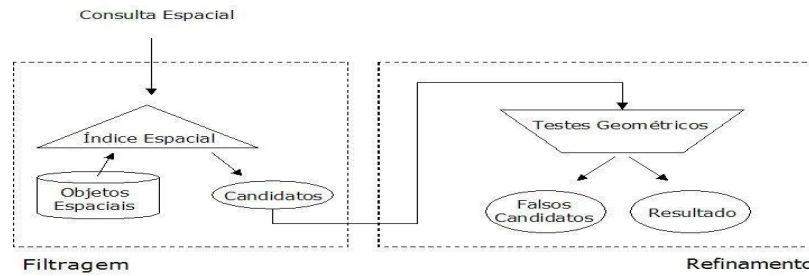


Figura 8: Processamento de consultas espaciais
Fonte: DAVIS Jr.(a) e QUEIROZ (2005).

Exemplos de métodos de indexação são: *Quad-tree* e *R-tree*. Estes métodos de indexação são alguns dos mais utilizados pelos SIGs e SGBDs com extensão espacial. É importante salientar que alguns dos SGBDEs podem trabalhar com mais de um mecanismo de indexação.

2.4.5 Padronização em Banco de Dados

2.4.5.1 Padrão SQL

Em processamento de banco de dados o usuário utiliza o SQL para interagir com o computador. Originalmente projetada e desenvolvida para recuperar dados de bancos de dados relacionais SQL é hoje a linguagem padrão para consulta e gerenciamento tanto de bancos de dados relacionais como objeto-relacional. Desde a sua adoção pelo *American National Standard Institute* (ANSI) como uma linguagem padrão de banco de dados em 1986, SQL foi formalmente aceita como um padrão internacional pela *International Organization for Standardisation* (ISO) e a *International Electrotechnical Commission* (IEC). (YEUNG e HALL, 2007).

SQL é uma linguagem de computador não-procedural, pois ela não tem declarações IF para condições de teste, nem contém declarações WHILE, FOR, GOTO CASE para a estruturação do programa e controle de fluxo. É melhor descrita como uma sub-linguagem de banco de dados usada especificamente para criar declarações para consulta e gerenciamento de banco de dados. (YEUNG e HALL, 2007).

A SQL é a linguagem utilizada para definir e manipular dados em um banco de dados relacional e objeto-relacional. Ela baseia-se na combinação de construtores em álgebra e do cálculo relacional e não é uma linguagem computacionalmente completa. O Cálculo Relacional de Tupla é uma Linguagem de Consulta formal declarativa, não-procedural,

baseada num ramo da lógica matemática chamado Cálculo de Predicado. Um predicado é uma declaração que avalia uma expressão e provê uma resposta verdadeira ou falsa baseada numa condição. Mais detalhes sobre a Álgebra Relacional e o Cálculo Relacional de Tupla podem ser obtidos em SILBERSCHATZ, KORTH e SUDARSHAN (1999), e ELMASRI e NAVATHE (2000).

Desde que foi definido pela ANSI este primeiro padrão ficou conhecido como SQL-86. Após algumas melhorias realizadas, um novo padrão foi adotado sendo chamado então de SQL-89. Já em 1992 o modelo foi definido. Este novo modelo ficou conhecido como SQL-92 (SQL2). Estas mudanças foram finalizadas em 1999 (SQL-99 ou SQL3) com alterações significativas e uma delas relacionada com a definição de padrões para banco de dados Objeto-Relacionais. De acordo com FERREIRA et al. (2005) a linguagem SQL é adotada pela maioria dos SGBD-R e SGBD-OR comerciais.

SQL é formada basicamente por duas sub-linguagens, na Tabela 1 são apresentados alguns comandos em SQL:

a) Linguagem de definição de dados (SQL DDL - *Data Definition Language*): fornece comandos para definir e modificar esquemas de tabelas, remover tabelas, criar índices e definir restrições de integridade.

b) Linguagem de manipulação de dados (SQL DML - *Data Manipulation Language*): fornece comandos para consultar, inserir, modificar e remover dados no banco de dados.

Tabela 1 – Comandos em SQL
Fonte: FERREIRA et al., 2005

Comando	Descrição	Tipo
select	Recupera dados de uma ou mais tabelas	DML
insert update delete	Servem para incluir, alterar e eliminar registros de uma tabela, respectivamente	DML
Commit rollback	Responsáveis pelo controle de transações, permitem que o usuário desfça (rollback) ou confirme (commit) alterações em tabelas	DML
create alter drop	Permitem a criação, modificação e remoção de objetos, incluindo tabelas	DDL

A Figura 9 mostra um exemplo de utilização do comando *Select* da Linguagem SQL.

SELECT NomeCli, EndCli

FROM Clientes C

WHERE C.Multa <> 0

CodCli	NomeCli	EndCli	Multa
23421	Daniel	Rua Alberto Xavier	0
24519	Pâmela	Av. Agamenon Magalhães	10
21953	Otávio	Rua Des. Mario Almeida	0

Figura 9: Comando *SELECT* em uma tabela relacional “Mostre um relatório com o nome dos clientes e seus endereços que já atrasaram a devolução de filmes em uma locadora.”

A SQL92 é a versão da SQL mais atual para banco de dados puramente relacionais. O padrão chamado SQL3 inclui, além de outras características, tipos de dados definidos pelo usuário e facilidades normalmente associadas a gerenciamento de dados orientados a objetos o que o torna uma extensão do padrão SQL para suportar bancos de dados objeto-relacionais.

Para as consultas espaciais FERREIRA et al. (2005) analisa que as extensões espaciais devem considerar dois pontos básicos:

- embora seja possível estender SQL com operadores espaciais, a semântica destes operadores deve ser formalmente definida;
- embora seja possível estender SQL para incluir controle de apresentação de geometrias, aconselha-se projetar uma linguagem separada para lidar com esta questão .

Os tipos de consultas espaciais que se pode aplicar em uma base de dados espacial são divididos, conforme FERREIRA et al. (2005), em 2 tipos mais importantes:

- *seleção espacial*: dado um conjunto de objetos espaciais D e um predicado de seleção espacial ρ sobre atributos espaciais dos objetos em D , determine todos os objetos em D cujas geometrias satisfazem ρ ;
- *junção espacial*: dados dois conjuntos de dados espaciais, D e D' , e um predicado de seleção espacial θ , determine todos os pares $(d,d') \in D \times D'$ cujas geometrias satisfazem θ .

Os seguintes casos particulares pertencentes à seleção espacial podem ser descritos:

- *seleção por ponto*: dado um ponto P e um conjunto de objetos espaciais D , determine todos os objetos em D cujas geometrias contêm P .
- *seleção por região*: dada uma região R e um conjunto de objetos espaciais D , determine todos os objetos em D cujas geometrias estão contidos em R .
- *seleção por janela*: dado um retângulo R com os lados paralelos aos eixos e um conjunto de objetos espaciais D , determine todos os objetos em D cujas geometrias estão contidos em R .

Em seguida são explicados padrões SQL para extensões que incluem suporte a consultas espaciais na linguagem SQL, incluindo os padrões publicados pela OGC e ISO,

chamados SFSSQL e SQL/MM Spatial, respectivamente.

2.4.5.2 Padrão SFSSQL

A especificação *OpenGIS Simple Features Specification For SQL – SFSSQL* (OGC, 1999) baseada no padrão SQL (*Structured Query Language*), produzida pelo OGC (*Open Geospatial Consortium*), especifica um conjunto de tipos de geometrias vetoriais, operações topológicas e operações métricas.

O Open Geospatial Consortium (OGC, 2005) é uma organização internacional sem fins lucrativo e formada por 271 associados, entre os quais: empresas, agências governamentais, universidades e institutos de pesquisa. Foi criada com o objetivo de desenvolver padrões consensuais envolvendo serviços geoespaciais e de localização. Como atribuições esse consórcio promove o desenvolvimento de tecnologias que facilitam a interoperabilidade entre sistemas estabelecendo padrões para a distribuição e interligações de dados de diversas fontes, facilitando a construção de Sistemas de Geoinformação. Os produtos do trabalho do OGC são apresentados sob forma de especificações de interfaces e padrões de intercâmbio de dados (DAVIS Jr.(c) et al., 2005). A especificação SFSSQL apresentada nesta subseção é um desses produtos.

OGC define um padrão que permite a composição de atributos espaciais e convencionais. De acordo com FERREIRA et al. (2005) a SFSSQL especifica também um esquema de tabelas para metadados das informações espaciais que introduz o conceito de "tabela com feições" (*FEATURE TABLE*) para representação dos dados geográficos.

A SFSSQL define um modelo chamado de "modelo de geometria". O modelo de geometria consiste de uma hierarquia de classe, que é mostrado na Figura 10. Este diagrama é o mesmo tanto para o modelo da SQL-92 quanto para o da SQL-92 que estende os tipos SQL utilizando tipos abstratos de dados específicos para os Tipos Geométricos. O modelo de geometria é um modelo abstrato. Ele é usado para definir a relação entre as diversas classes e estabelecer as regras de herança para os métodos de trabalho nas instâncias das classes e subclasses. Por exemplo, o método *Area* é definido para a classe *Surface* e está disponível para todas as instâncias de *Surface*, *Polygon*, e outras sub-classes, enquanto o método *ExteriorRing* é apenas definido na subclasse *Polygon* e, portanto, não pode ser usado para instâncias arbitrárias da classe *Surface*.

Alguns tipos são abstratos como: *Curve*, *Surface*, *MultiSurface* e *MultiCurve*. Um tipo especial é a *GeometryCollection*, que pode ser composta por mais de um tipo de

geometria (tipo heterogêneo). Os outros são tipos básicos, como Point, LineString e Polygon, que podem formar tipos de coleções homogêneas como MultiPoint, MultiLineString e MultiPolygon, respectivamente. Cada um destes tipos possui uma série de atributos, métodos e definições que são apresentadas na especificação (FERREIRA et al., 2005).

A seguir considera-se, por exemplo, a consulta espacial:

Q. Selecione os municípios que fazem fronteira com o município de Recife.

Esta consulta pode ser expressa em SFSSQL da seguinte forma:

```
SELECT M1.name
FROM Municipio M1, Municipio M2
WHERE Touch(M1.location, M2.location)=1
AND M2.Name = 'Recife'
```

Observa-se que a sintaxe da SQL para esta consulta espacial, é muito semelhante á apresentada na Figura 9. As localizações dos municípios são armazenadas como geometrias em uma coluna espacial e devido à característica da Arquitetura Integrada, essas localizações são armazenadas na mesma tabela dos dados de municípios, tal como o nome do município. Isso se deve ao fato da SQL-92 estender os tipos SQL utilizando tipos abstratos de dados específicos para os Tipos Geométricos.

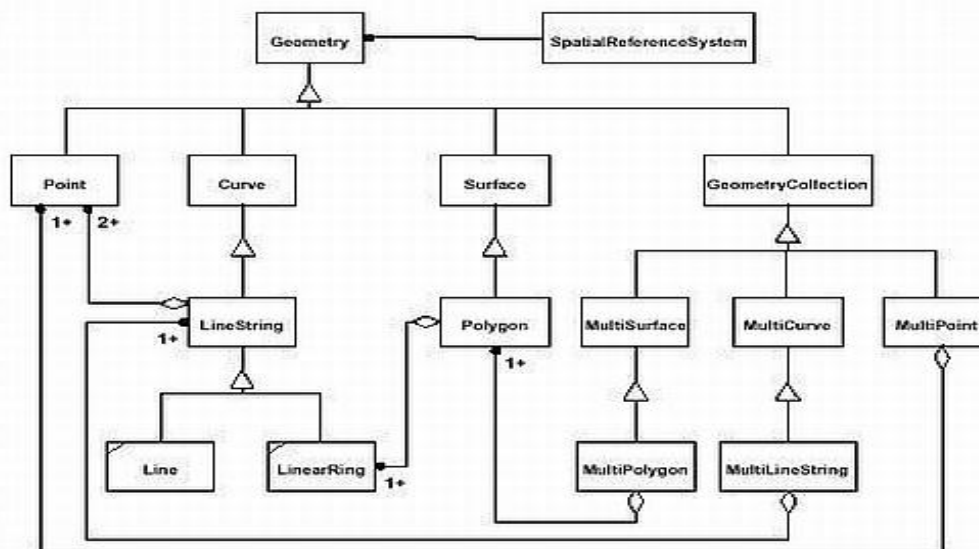


Figura 10: Hierarquia de tipos de geometrias da SFSSQL

Fonte: FERREIRA et al., 2005.

O esquema de metadados da especificação SFSSQL representa os dados geoespaciais em tabelas com atributos dos tipos de geometrias anteriormente descritos:

SPATIAL_REF_SYS, tabela armazenando dados sobre cada sistema de referenciamento espacial (SRS) utilizado no banco de dados. GEOMETRY_COLUMNS, tabela de metadados para as colunas geométricas das tabelas com feições (feature tables).

Cada atributo do tipo Geometry (ou de seus sub-tipos), de cada tabela com feições, deve estar associado a um SRS, de tal forma que seja possível determinar o sistema de referenciamento espacial utilizado para representar cada geometria armazenada na tabela. Isso é essencial para que os métodos possam verificar a compatibilidade dos sistemas de referenciamento espacial utilizados pelas geometrias.

2.4.5.3 Padrão SQL/MM Spatial

O padrão ISO/IEC 13249-3 SQL/MM Part 3: Spatial (ISO, 2004) define tipos estruturados da SQL:1999 definidos pelo usuário e métodos associados para prover a capacidade de suportar tais aplicações (MELTON e EISENBERG, 2001).

O padrão ISO/IEC 13249 SQL/MM (MM para *MultiMedia*) representa o esforço para padronizar as extensões de multimídia e pacotes de aplicativos específicos em SQL. SQL é estendida para gerenciar dados como textos, imagens, dados geográficos, ou para realizar mineração de dados. O padrão é agrupado em várias partes bastante independentes entre si. (STOLZER, 2003).

ISO/IEC 13249-3 SQL/MM Part 3: Spatial (ISO, 2004) é o padrão internacional que define a forma de armazenar, recuperar e processar dados espaciais utilizando SQL. Ele define como dados espaciais devem ser representados como valores, e quais funções estão disponíveis para converter, comparar, e processar estes dados de várias maneiras. O padrão SQL/MM Spatial é dividido em cláusulas. As cláusulas de 5 a 9 descrevem os tipos de geometria e os métodos providos para cada tipo. O esquema de informação, baseado em um Esquema de Definição é definido na cláusula 14. As demais cláusulas, que não são citadas neste trabalho, explicam os conceitos espaciais subjacentes, o manuseio de direção e ângulo, os códigos de Estados e regras de conformidade para os produtos que implementam o padrão (STOLZER, 2003).

Pela sua própria natureza, os dados espaciais representam muitas vezes 2-dimensões e 3-dimensões. SQL/MM Spatial suporta atualmente dados de 0-dimensão (ponto), 1-dimensão (linha), e 2-dimensões (forma "plana"). O SQL/MM Spatial também modela sistemas de referenciamento espacial (SRS). Revisões futuras poderão suportar 3-dimensões

(Forma volumétrica) e, possivelmente, dados de dimensões ainda maiores (MELTON e EISENBERG, 2001).

O SQL/MM Spatial define na hierarquia de tipos um tipo mais generalizado (ou seja, seu supertipo máximo) chamado de `ST_Geometry`. Esse tipo não é instanciável (Significando que nenhuma instância dele pode ser criada – O SQL/MM Spatial definiu menos de meia dúzia de tais tipos), mas no padrão tem um número (cerca de uma dúzia) de subtipos que são instanciáveis, como `ST_Point` e `ST_MultiPolygon` (MELTON e EISENBERG, 2001).

O padrão foi originalmente derivado do OpenGIS Simple Features Specification for SQL (OGC, 1999), também publicado no ano de 1999 como a versão 1.1 pelo Consórcio OpenGIS (OGC).

A Figura 11 mostra a hierarquia de tipos geométricos de SQL/MM Spatial, adaptada da hierarquia definida pela OGC, onde os tipos sombreados não são instanciáveis. (STOLZE, 2003). Conforme resume FERREIRA et al. (2005), a hierarquia de tipos geométricos da SQL/MM Spatial difere, porém, da hierarquia da OGC em alguns pontos:

- Adota `ST_LineString` em substituição a `Line` e `LinearRing`;
- Oferece arcos circulares e regiões cujas fronteiras são arcos circulares;
- Omite a indicação de que tipos são agregações de outros tipos.

Observando a Figura 11, não parece óbvio que uma geometria do tipo `ST_MultiPoint` seja uma agregação de geometrias do tipo `ST_Point`.

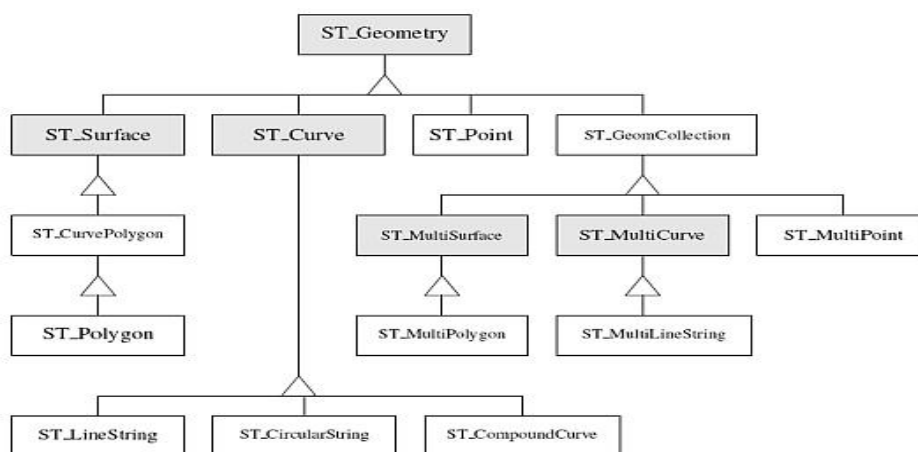


Figura 11: Hierarquia de tipos do SQL/MM Spatial

Fonte: STOLZE, 2003.

Os métodos que implementam as operações espaciais no padrão SQL/MM Spatial, de acordo com FERREIRA et al. (2005) e STOLZE (2003) podem ser agrupados em

quatro categorias (conforme operações espaciais definidas na subseção 2.4.3) :

- métodos para conversão entre geometrias e formatos de dados externos. O SQL/MM Spatial define três formatos de dados externos: well-known text representation (WKT), well-known binary representation (WKB) e geography markup language (GML). O padrão define métodos construtores que podem manusear representações WKT e WKB, contudo não permite manuseio de GML com um construtor. Um conjunto de funções permite ao usuário construir uma geometria usando um dos formatos de dados externo. Os três métodos *ST_AsText*, *ST_AsBinary* e *ST_AsGML* são providos pelo padrão para converção de uma geometria ao respectivo formato de dados externo;
- métodos para recuperar propriedades métricas de geometrias. Ex.: *ST_IsValid* (testa se uma geometria é válida), *ST_IsEmpty* (testa se uma geometria é vazia), *ST_Length* (retorna o tamanho de uma *linestring* ou *mult-linestring*), etc.
- métodos para comparar duas geometrias implementando relacionamentos topológicos. Ex.: *ST_Equals* (testa a igualdade de duas geometrias), *ST_Disjoint* (testa se duas geometrias não se interseptam), *ST_Touches* (testa se duas geometrias tocam em suas bordas, mas não interseptam em seus interiores), etc;
- métodos gerando geometrias a partir de outras. Ex.: *ST_Buffer* (gera um buffer em uma distância específica em torno de uma dada geometria); *ST_ConvexHull* (processa o convex hull para uma geometria), etc.

A Parte 3 da especificação do SQL/MM Spatial, segundo FERREIRA et al. (2005), define um esquema de metadados, semelhante ao esquema definido para o SFSSQL. O esquema de metadados compreende as seguintes tabelas:

- *ST_GEOMETRY_COLUMNS*, armazena metadados descrevendo as colunas geométricas das tabelas do banco de dados (idêntica à tabela *GEOMETRY_COLUMNS* da SFSSQL, definida pela OGC);
- *ST_SPATIAL_REFERENCE_SYSTEMS*, armazenando dados sobre cada sistema de referenciamento espacial (SRS) utilizado no banco de dados (idêntica à tabela *SPATIAL_REF_SYS* da SFSSQL);
- *ST_UNITS_OF_MEASURE*, armazena as unidades de medida utilizadas no banco de dados.
- *ST_SIZINGS*, semelhante à tabela *SIZINGS* do SQL99, define as meta-variáveis, e seus valores, específicas para a componente especial do banco de dados.

2.4.6 Extensões Espaciais de SGBD

Utilizam objetos complexos oferecidos por gerenciadores objeto-relacionais com: tipos de dados espaciais (ponto, linha, polígono, etc.); operadores e funções para manipular esses tipos de dados; métodos de acesso eficientes para esses tipos de dados.

Cada instância espacial tem um SRID (spatial reference identifier). O SRID corresponde a um sistema de referência espacial baseado no elipsóide específico usado para mapeamento de terra plana ou de terra redonda. Uma coluna espacial pode conter objetos com SRIDs diferentes. No entanto apenas instâncias espaciais com o mesmo SRID podem ser usadas ao executar operações com métodos de dados espaciais.

Atualmente, existem várias extensões comerciais e de código aberto disponíveis para tratar de dados geográficos no formato vetorial: Oracle Spatial (Ravada and Sharma 1999), IBM DB2 Spatial Extender e Informix. No universo do software de código fonte aberto e gratuito, existe por exemplo, a extensão geográfica baseada no PostgreSQL, chamada de PostGIS (Ramsey 2002). Todas essas extensões baseiam-se nas especificações do OGC (Consortium 1995), porém, apresentam variações relevantes entre os modelos de dados, semântica dos operadores espaciais, mecanismos de indexação e esquema e sintaxe da SQL estendida com tipos espaciais.

As extensões espaciais tratam somente objetos espaciais cuja componente espacial seja uma geometria vetorial. Os dados matriciais são armazenados em campos binários longos (BLOBs) que não se utilizam dos recursos oferecidos pelas extensões espaciais. Este trabalho abrange apenas o tratamento das representações de dados vetoriais nas extensões espaciais estudadas. A seguir, são apresentadas as características e funcionalidades do Oracle Spatial e do PostgreSQL/PostGIS.

2.4.6.1 Oracle Spatial

A extensão espacial Oracle Spatial (MURRAY, 2003 apud QUEIROZ e FERREIRA, 2005) foi desenvolvida para o modelo objeto-relacional do SGBD Oracle e é baseada nas especificações do OGC. O SGBD Oracle disponibiliza o Oracle Spatial desde a versão 8i à atual 11g. O Oracle disponibiliza a extensão espacial em todas as suas edições entretanto na *Enterprise Edition* a solução é mais completa incluindo *Raster*, *Geocoding*, *Mineração de dados*, *roteamento*, dentre outros.

O Oracle Spatial fornece um esquema SQL e funções que facilitam o armazenamento, recuperação, atualização e consulta de coleções de feições espaciais em um banco de dados Oracle. O Oracle Spatial é formado pelos seguintes componentes (ORACLE, 2010):

- Um esquema (chamado MDSYS) que define o armazenamento, syntax e semântica dos tipos de dados espaciais suportados;
- Mecanismo de indexação espacial;
- Um conjunto de operadores, funções e procedures para realizar consultas, junção espacial e outras operações de análise espacial;
- Funções e procedures para operações de utilitários e operações de performance (tuning);
- Modelo de dados de topologia para trabalhar com dados sobre nós, arestas, e faces em uma topologia;
- Modelo de dados de rede para representação objetos que são modelados como nós e ligações em uma rede;
- *GeoRaster* que é um recurso para permitir armazenamento, indexação, consulta, análise e distribuição de imagem e dados de grade e seus metadados associados;

O modelo de dados do Oracle Spatial consiste em uma arquitetura hierárquica de elementos, geometrias e planos de informação (*layers*) (QUEIROZ e FERREIRA, 2005).

Um plano de informação é formado por um conjunto de geometrias. As geometrias por sua vez, podem ser formadas por um conjunto de elementos. E os elementos são formados por tipos geométricos primitivos como ponto, cadeia de linhas, polígonos, etc. Os tipos geométricos disponíveis estão ilustrados na Figura 12.

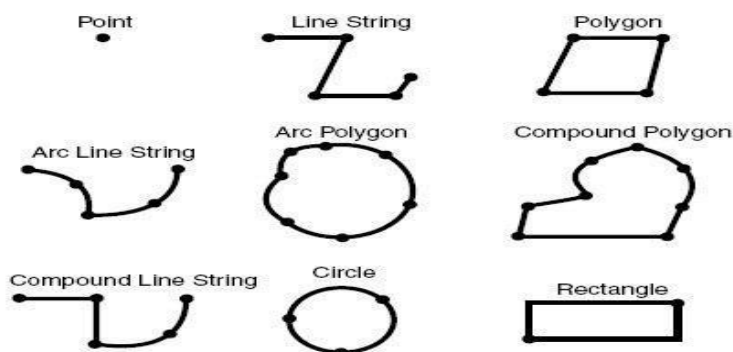


Figura 12: Tipos espaciais primitivos do Oracle Spatial
Fonte: QUEIROZ e FERREIRA, 2005.

O Oracle possui um conjunto de funções de análise espacial maior do que o de seus concorrentes. Também possui o maior número de geometrias do padrão OGC, segundo PROJETO SMARTVIEW(a) (2008), com suporte a feições do tipo ponto, linha, polígono, multilinha, multipolígono, arc linha, arc polígono, círculo, etc. Além de uma biblioteca mais completa para trabalhar com dados 3D. É possível também especificar um nível de tolerância na distância entre dois pontos, o que será levado em consideração quando o SGBD calcular quando duas geometrias se tocam, ou são iguais.

O Oracle Spatial, de acordo com o modelo objeto-relacional, define um tipo de objeto para armazenar sua geometria, chamado SDO_GEOMETRY. Este objeto contém a sua geometria, o par de coordenadas e informações sobre seu tipo e projeção. Na tabela espacial existem atributos alfanuméricos da geometria que são definidos com tipos convencionais (number, varchar2, etc.) e outro atributo com a geometria, que é uma coluna do tipo SDO_GEOMETRY.

De acordo com QUEIROZ e FERREIRA (2005) um objeto SDO_GEOMETRY é composto pelos seguintes atributos:

- SDO_GTYPE: formado por quatro números, onde 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.

Estes são operadores para objetos simples no Oracle Spatial. Existem várias outras funções para trabalhar com objetos 3D, cálculo de rota, funções para dados *raster*, etc.

Em relação à indexação de colunas espaciais, o Oracle Spatial utiliza a técnica *R-Tree* e pode indexar geometrias com até 4 dimensões. À medida que o usuário realiza operações de *insert* e *delete* no SGBD a indexação que foi criada usando o *R-Tree* pode ter seu desempenho comprometido. Porém, o Oracle Spatial possui uma função chamada *Quality_Degradation* que informa o quanto a indexação está comprometida. Com base nessa informação, o DBA pode reconstruir o índice da coluna espacial quando o nível de degradação estiver acima do permitido.

De acordo com QUEIROZ e FERREIRA (2005) para realizar consultas espaciais são utilizados os operadores:

- SDO_FILTER: implemente o primeiro filtro do modelo de consulta, ou seja, verifica se os MBRs das geometrias têm alguma interação entre si.
- SDO_RELATE: consulta relações topológicas entre objetos geográficos. Recebe como parâmetro o tipo da relação a ser computada, na qual pode ser: equal, disjoint, touch, inside, overlapdyintersect, overlapbdydisjoint, anyinteract, contains, on, covers e coveredby;
- SDO_WITHIN_DISTANCE: verifica se duas geometrias estão a uma determinada distância passada por parâmetro;
- SDO_NN: Identifica *n* vizinhos mais próximos de um objeto;

Estes são operadores para objetos simples no Oracle Spatial. Existem várias outras funções para trabalhar com objetos 3D, cálculo de rota, funções para dados *raster*, etc.

O Oracle possui funções que auxiliam nas consultas como: SDO_BUFFER, gera uma nova geometria em torno de outra de acordo com o parâmetro de distância passado; SDO_AREA E SDO_LENGTH calculam a área e o perímetro de um objeto; SDO_DISTANCE calcula a distância entre duas geometrias; SDO_INTERSECTION, SDO_UNION, SDO_DIFFERENCE geram uma nova geometria a partir da interseção, união e diferença, respectivamente entre outras duas.

Os operadores são usados na cláusula WHERE e utilizam indexação espacial. Só podem ser usados em colunas espaciais indexadas. As funções, são definidas como

subprogramas em SQL, e utilizadas na cláusula WHERE ou em subconsultas, podendo ser executadas sobre colunas espaciais não indexadas. Devido ao fato dos operadores sempre explorarem a indexação, é recomendável usá-los, em lugar de funções, quando possível.

Ainda conforme PROJETO SMARTVIEW(a) (2008), o Oracle é líder no mercado em segurança, desempenho, escalabilidade e gerência de dados espaciais armazenados em banco de dados.

2.4.6.2 PostGreSQL/PostGis

O PostGreSQL é um SGBD objeto-relacional (SGDB-OR), oriundo do projeto chamado de PostGres, criado pelo professor Michael Stonebraker, na Universidade da Califórnia em Berkeley. O PostGres foi desenvolvido entre 1986 e 1994 (QUEIROZ e FERREIRA, 2005).

Em 1995, o PostGres passou por alterações, incorporando alguns conceitos de SQL. A partir de 1996, o SGDB sofreu várias alterações, devido ser de código aberto, seu código fonte foi disponibilizado na WEB (<http://www.postgresql.org>) e daí em diante um grupo de desenvolvedores continuou com a manutenção e aperfeiçoamento do SGBD que passou a se chamar PostGreSQL (QUEIROZ e FERREIRA, 2005).

O PostGreSQL possui suporte nativo a dados espaciais, porém seus recursos originais não são muito abrangentes e eficientes. Para suprir essas deficiências a empresa Refrations Research Inc. (<http://www.refrations.net>) desenvolveu e mantém a extensão espacial PostGIS.

O PostGIS é um projeto de código aberto e gratuito sob licença GPL (*GNU General Public License*). É uma expansão de software que oferece ao PostGreSQL o suporte a objetos espaciais de acordo com a especificação SFSSQL.

O PostGIS suporta os seguintes tipos de geometrias, embutidas na SQL do PostGreSQL, que seguem as especificações propostas pela SFSSQL da OGC: *Point*, *Linestring*, *Polygon*, *Multipoint*, *Multilinestring*, *Multipolygon*, *GeometryCollection*. A Figura 13 ilustra estes tipos de geometrias.

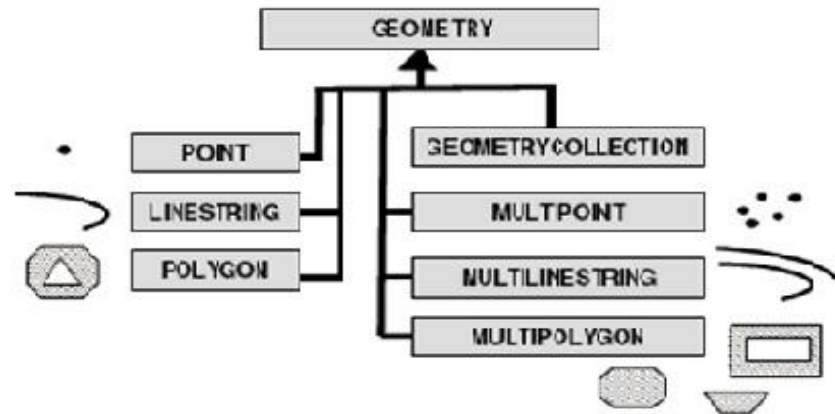


Figura 13: Tipos de dados espaciais do PostGIS
Fonte: QUEIROZ e FERREIRA, 2005.

A seguir é apresentado um exemplo em SQL que cria uma tabela e armazena os atributos e geometria:

```
CREATE TABLE CASAS (NUM_CASA INT4,END VARCHAR(25));  
SELECT AddGeometryColumn('BD','CASAS','geom',29101,'POINT',2);
```

Quando é criada uma tabela com coluna espacial, deve ser informado o tipo da coluna (*Point*, *Poligon*, etc) e o SRID da mesma. Pelo exemplo, observa-se que a criação da tabela com tipo espacial faz-se em dois passos. No primeiro passo define-se os atributos alfanuméricos e no segundo utiliza-se a função *AddGeometryColumn* para adicionar a coluna com o tipo espacial. Uma tabela de metadados chamada “*geometry_columns*” é preenchida através da função *AddGeometryColumn* que consta nas especificações da SFSSQL e seus parâmetros são: nome do banco de dados, nome da tabela que irá conter a coluna espacial, nome da coluna espacial, sistema de coordenadas em que se encontra as geometrias da tabela, tipo da coluna espacial e dimensão das coordenadas dos dados.

As tabelas de metadados do PostGIS seguem as especificações da SFSSQL e são representadas pelas tabelas *spatial_ref_sys* (tabela de metadado do sistema de coordenadas) e *geometry_columns* (tabela de metadados das tabelas com colunas espaciais).

Um comando SQL INSERT pode ser usado para inserir as informações como no exemplo a seguir:

```
INSERT INTO casas (num_casa,end,geom)
```

```
VALUES(75,'ED. ALAMEDA',
GeomFromText('POINT(-34.917427 -8.029719)',29101));
```

Com um comando SQL SELECT pode-se recuperar informações em cada tabela como no exemplo a seguir:

```
SELECT descrição, AsText(geom) geometria FROM casas c
WHERE c.num_casa = 50;
```

Observa-se que nos exemplos acima foram empregadas as funções de entrada *GeomFromText* e a de saída *AsText* para representação textual (WKT). O PostGIS também utiliza o formato binário (WKB) de representação cujas funções de entrada e saída são respectivamente *GeomFromWKB* e *AsBinary*. Os formatos WKT e WKB constam nas especificações SFSQL da OGC.

Quando o SGBD trabalha com um grande volume de informações é possível tornar as buscas mais eficientes através do uso de indexação. Conforme **PROJETO SMARTVIEW(a) (2008)** o PostgreSQL juntamente com o PostGIS, oferecem três tipos de indexação de dados: *B-Tree* para dados não espaciais, *R-Tree* e GiST (*Generalized Search Trees*). As colunas espaciais podem ser indexadas através de uma *R-Tree* construída no em cima do GiST tornando a consulta a dados espaciais mais rápida. *R-Tree* indexa as geometrias em retângulos, sub-retângulos e su-sub retângulos. Porém, no PostgreSQL, uma forma mais robusta que índice *R-Tree* é o índice GiST. O GiST organiza os dados em “objetos que estão ao lado”, “objetos que sobrepõem”, “objetos que estão dentro”, etc.

Vários operadores espaciais são disponibilizados na extensão PostGIS, entre alguns deles cita-se (QUEIROZ e FERRREIRA, 2005):

- Operadores topológicos conforme a Matriz de 9-Interseções DE: `equals(geometry, geometry)`; `disjoint(geometry, geometry)`; `intersects(geometry, geometry)`; `touches(geometry, geometry)`; `crosses(geometry, geometry)`; `within(geometry, geometry)`; `overlaps(geometry, geometry)`; `contains(geometry, geometry)`; `relate(geometry, geometry)`: retorna a matriz de intersecção;
- Operador de construção de mapas de distância: `buffer(geometry, double, [integer])`;
- Operador para construção do Fecho Convexo: `convexhull(geometry)`;
- Operadores de conjunto: `intersection(geometry, geometry)`; `geomUnion(geometry, geometry)`; `symdifference(geometry, geometry)`; `difference(geometry, geometry)`;

- Operadores Métricos: `distance(geometry,geometry)`; `area(geometry)`;
- Centróide de geometrias: `Centroid(geometry)` ;
- Validação (verifica se a geometria possui auto-interseções): `isSimple(geometry)`;

3 ESTUDO COMPARATIVO

O estudo comparativo que será apresentado neste capítulo tem como base pesquisas realizadas sobre as extensões espaciais comparadas e elaboração de cenários que ilustrarão o estudo, possibilitando uma análise de diversos aspectos dos métodos espaciais. É importante salientar que o escopo do trabalho limita-se à análise das funcionalidades espaciais das extensões comparadas. Não serão explorados aqui aspectos de instalação e estabilidade.

3.1 Comparação entre as Extensões Espaciais *Oracle Spatial* e *PostGIS*

Nesta seção é apresentado um comparativo que mostra as principais características e funcionalidades espaciais de cada SGBD. A comparação é realizada tendo como balizadores as SFSSQL (OGC, 1995) e SQL/MM Spatial (ISO, 2004) fazendo-se uma avaliação principalmente no que tange aos tipos de dados, operadores e funções espaciais.

O Oracle Spatial possui o maior número de geometrias do padrão OGC com suporte a feições do tipo, linha, polígono, multilinha, multipolígono, arc linha, arc polígono, círculo, etc. Possui também uma biblioteca mais completa para trabalhar com dados 3D (**PROJETO SMARTVIEW(a), 2008**).

O PostGIS segue o padrão OGC e suporta as geometrias: ponto, linha, polígono, multiponto, multilinha, multipolígono e coleção de geometria, baseados na especificação SFSSQL (Simple Features for SQL). O PostGis oferece suporte a coordenadas 2D, 3D e 4D sendo os dois últimos com suporte apenas para armazenamento. Na coluna espacial do PostGis, deve ser informado o tipo da coluna (*Point*, *Linestring*, *Polygon*, etc) e o SRID (*spatial reference identifier*) da mesma. Cada objeto espacial é uma instância espacial que possui o seu SRID. Apenas colunas com o mesmo SRID (mesma unidade de medida, datum e projeção usada para determinar as coordenadas do objeto) podem ser usadas ao executar operações com métodos espaciais.

O PostGIS possui um grande número de funções disponíveis para análises espaciais e topológicas como por exemplo operadores topológicos, conforme a matriz de 9-interseções DE (referenciado na SFSSQL do padrão OGC). O Oracle Spatial possui mais de 400 funções espaciais. As funções de relacionamento topológicos do Oracle Spatial são baseadas no modelo de 9-interseções. Ambas as extensões podem calcular centróides, gerar novas geometrias com operações de *buffers*, união, interseção, entre outras conforme

especificado nos padrões SFSSQL e SQL/M Spatial. Também possuem uma função para realizar transformação entre sistemas de referências em conformidade com o padrão SQL/MM Spatial.

Essas duas extensões espaciais possuem tabelas de metadados das colunas espaciais. O PostGIS implementa a tabela de metadados conforme o padrão SFSSQL enquanto o Oracle Spatial implementa diferente do padrão.

O Oracle Spatial e o PostGIS aceitam vários formatos para carga de dados espaciais como WKT, WKB, GML, KML e ESRI SHP (*shapefile*). O WKT e WKB estão em SFSSQL e SQL/MM Spatial. O GML está em SQL/MM Spatial enquanto o KML é um padrão mantido pelo Open Geospatial Consortium, Inc. (OGC).

O Oracle Spatial apresenta uma total discordância em relação à nomenclatura original das funções e procedimentos da especificação SFSSQL. Do ponto de vista da especificação original, esse fato dificulta a identificação dessas funções bem como compromete a interoperabilidade com outros produtos que estão conforme o padrão.

Com relação à indexação espacial, o PostGIS usa índice de *R-Tree* implementado em cima de GiST (*Generalized Search Trees*) para indexação de dados espaciais, tornando a consulta a esses dados mais rápida. O Oracle Spatial utiliza a técnica *R-Tree* que pode indexar geometrias até 4 dimensões. O Oracle Spatial também pode utilizar a técnica de indexação *Quadtree*, porém é recomendado pela Oracle, fortemente, o uso de *R-Tree* ao invés de *Quadtree*, por causa do seu desempenho (QUEIROZ e FERREIRA, 2005).

O Oracle Spatial, disponível na distribuição Oracle Enterprise Edition, representa uma solução mais completa incluindo *Raster*, *Geocoding*, Mineração de dados, roteamento, dentre outros.

O PostgreSQL/PostGIS se apresenta como uma alternativa ao Oracle Spatial oferecendo, para usuários e empresas que estão preocupados com economia de investimento em TI, a vantagem de ser um SGBD gratuito e de código aberto. É o SGBD que mais se assemelha ao padrão SFSSQL do OGC.

A tabela 2 a seguir apresenta um resumo dos estudos realizados para comparação das extensões.

Tabela 2 - Quadro Comparativo das extensões *Oracle Spatial* versus *PostGIS*
Fontes: (QUEIROZ e FERREIRA, 2005) , (BAPTISTA, 2010), (LOURENÇO, 2008)

Recursos	Oracle Spatial	PostGIS
Tipos Espaciais	Ponto, linha, polígono, multiponto, multilinha, multipolígono, coleção de geometria, arco-linha, arco-polígono, polígono composto, linha composta, círculo, retângulo; Segue a SFSSQL; Possui o maior número de geometrias do padrão OGC.	Ponto, linha, polígono, multiponto, multilinha, multipolígono, coleção de geometria, 2D, 3D E 4D, sendo os dois últimos com suporte apenas para armazenamento; Segue a SFSSQL.
Número de dimensões	4	4
Funções espaciais	Implementa diversas funções dentre elas funções especificadas pela OGC; Utiliza um mecanismo para dividir a consulta em duas etapas de forma a reduzir a complexidade; Possui mais de 400 funções espaciais.	Conta com um grande número de funções para análises espaciais/topológicas dentre elas funções especificadas pela OGC.
Nomenclatura das funções e procedimentos conforme Padrão OGC	Muita discordância com o padrão.	Pouca discordância com o padrão.
Indexação espacial	<i>R-tree</i> e <i>Quad-tree</i>	<i>R-tree</i> sobre a indexação GiST.
Formatos de dados espaciais suportados (WKT,GML, KML e ESRI SHP)	Sim	Sim
Importação de dados com o uso de <i>shapefile</i>	Sim (<i>shp2sdo</i> para carga de dados)	Sim (<i>shp2pgsql</i> para carga de dados)
Operadores topológicos	Matriz 9-Intersecções	Matriz 9-Intersecções DE
Operadores de conjuntos	Sim	Sim
Operador de <i>buffer regions</i>	Sim	Sim
Transformação entre sistemas de coordenadas	Sim	Sim
Tabelas de metadados das colunas geométricas	Sim (diferente do SFSSQL)	Sim (conforme SFSSQL)
Suporte ao armazenamento e manipulação de	Sim	Não

imagens (GeoRaster)		
Suporte ao armazenamento de redes (Grafos) e conectividade (Modelo Network)	Sim	Não
Transformação de endereço em coordenadas lat/long e inverso (GeoCoder)	Sim	Não
Suporte à funções analíticas em mineração de dados espaciais (Spatial Data Mining)	Sim	Não

3.2 Cenários para Consultas Espaciais

A fim de ser realizada uma análise prática das funções espaciais das duas extensões espaciais em estudo, é utilizada uma base de dados, composta por 4 tabelas, criada a partir de tabelas extraídas do banco de dados do SmartView (Anexo B).

O ambiente de instalação utilizado foi um microcomputador com a seguinte configuração de Hardware e Software:

- Processador: Intel Core i5, 1.4 GHz;
- Memória instalada (RAM): 4,00 GB;
- Tipo de sistema: Sistema Operacional de 32 Bits;
- Windows 7 Professional.

Para realização dos testes foram utilizadas as seguintes versões dos SGBDS e programas de interface para os respectivos bancos de dados:

- Oracle 11g Release 2 Enterprise Edition.
- PL/SQL Developer. Versão 8.0.1.1502;
- PostgreSQL. Versão 9.0; PostGIS. Versão 1.5.2-3.
- pgAdmin III. Versão 1.12.2 (28 de Janeiro de 2011) REL-1_12_2.

Para carga das tabelas foram utilizados aplicativos que carregam tabelas nos bancos de dados a partir de arquivos *shape*. Esses arquivos foram gerados a partir das tabelas existentes no banco de dados do SmartView. Para o Oracle Spatial foi utilizado o aplicativo *shp2sdo* e no caso do PostGIS foi utilizado o *shp2pgsql* que através desses aplicativos é possível extrair os dados dos arquivos *shape* e inserir no banco de dados. Um *shapefile* irá corresponder a uma tabela no banco de dados.

O Oracle Spatial e o PostGIS contam com um grande número de funções para análises espaciais/topológicas que estendem o próprio SQL (ORACLE ,2001) e (POSTGIS, 2010). Para este trabalho, foram escolhidos alguns cenários relacionados com o Sistema Elétrico da CHESF para realização de consultas espaciais nas extensões consideradas. Serão utilizadas algumas funções mais comuns nas tarefas de geoprocessamento tais como:

- ***Intersects:*** analisa se duas geometrias possuem alguma interseção e retorna verdadeiro (t, true) caso isto ocorra;
- ***Touches:*** analisa se duas geometrias possuem alguma interseção e retorna verdadeiro (t, true) caso isto ocorra;
- ***Crosses:*** analisa se duas geometrias se cruzam e retorna verdadeiro (t, true) caso isto ocorra;
- ***Within:*** analisa se uma geometria está contida na outra e retorna verdadeiro (t, true) caso isto ocorra;
- ***Contains:*** analisa se uma geometria contém a outra e retorna verdadeiro (t, true) caso isto ocorra;
- ***Disjoint:*** analisa se duas geometrias não interceptam espacialmente e retorna verdadeiro (t, true) caso isto ocorra;
- ***Overlaps:*** analisa se duas geometrias compartilham o espaço, mas uma não contém completamente a outra e retorna verdadeiro (t, true) caso isto ocorra;
- ***Distance:*** Calcula a distância de uma geometria até outra;
- ***Buffer:*** cria um buffer ao redor da geometria, aumentando sua área.
- ***Union:*** gera uma nova geometria da união entre outras duas.
- ***Centroid:*** retorna o centro geométrico da geometria.

Nos exemplos que serão apresentados, as informações de Subestações serão representadas por pontos, Linhas de Transmissão serão representadas por linhas e os Estados e Municípios da região Nordeste, onde a CHESF tem atuação, serão representados por polígonos.

As seguintes tabelas serão utilizadas nas consultas:

Ibge_estados_ne - Tabela que armazena as informações sobre os estados da região Nordeste onde a CHESF tem atuação.

Ibge_municipios_ne – Tabela que armazena as informações sobre os municípios da região Nordeste.

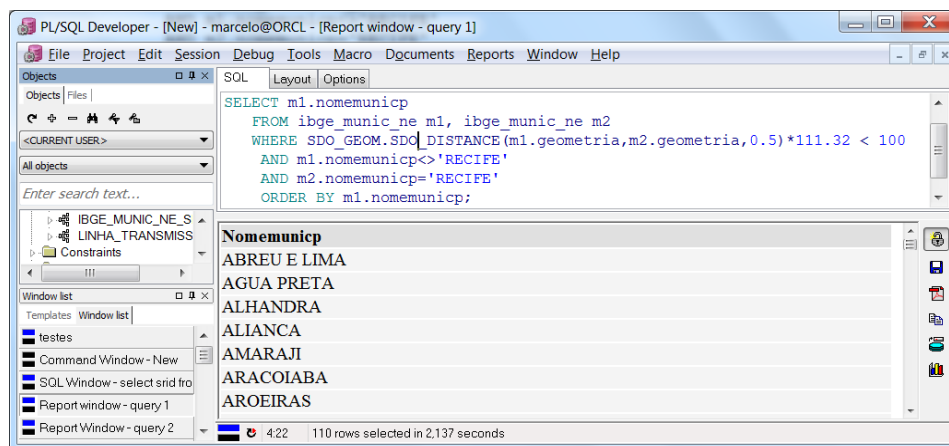
linha_transmissão - Tabela que armazena as informações sobre as linhas de transmissão cadastradas do sistema elétrico da CHESF.

Chesf_subestacoes - Tabela que armazena as informações sobre as subestações existentes no sistema elétrico da CHESF.

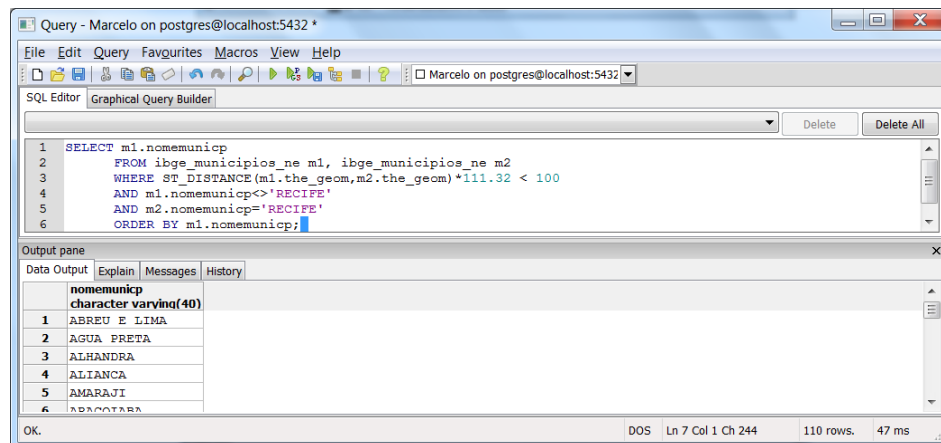
Os seguintes cenários envolvem as consultas em cada uma das extensões. Os scripts SQL para as consultas dos cenários se encontram no ANEXO A:

Cenário 1 – Encontrar todos os municípios que estejam a menos de 100 km do Recife;

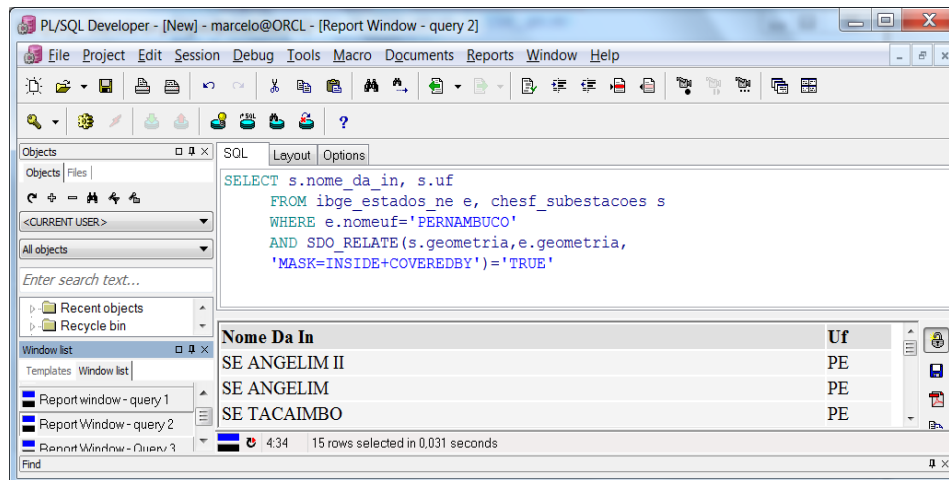
SQL em Oracle Spatial:



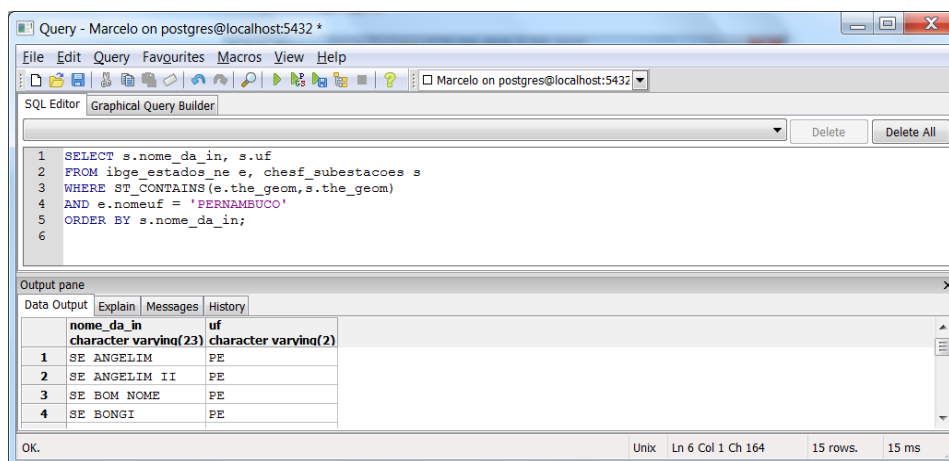
SQL em PostGIS:



Cenário 2 – Encontre todas as Subestações localizadas no estado de PE.

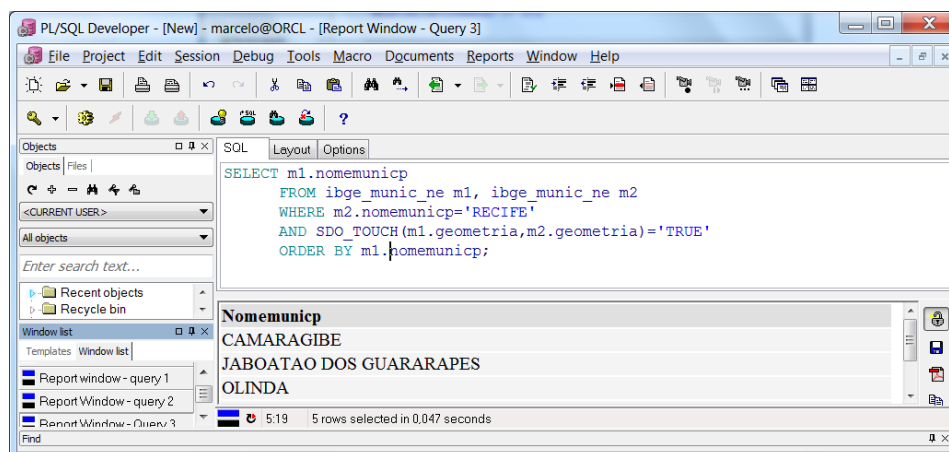


SQL em PostGIS:

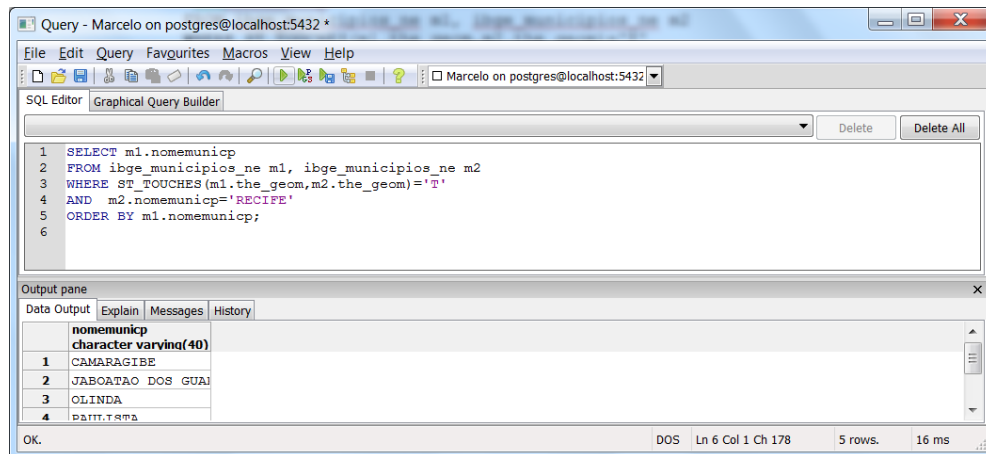


Cenário 3 – Mostre todos os municípios que são vizinhos de Recife.

SQL em Oracle Spatial:

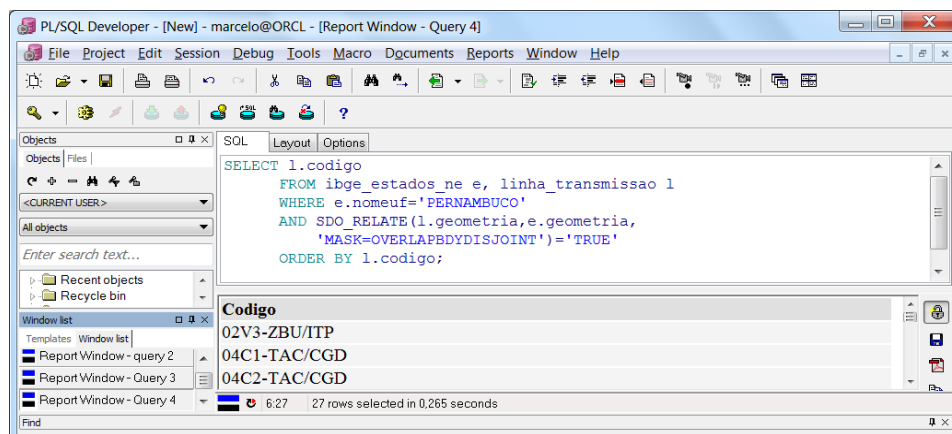


SQL em PostGIS:



Cenário 4 – Quais são as Linha de Transmissão que atravessam o estado de PE ?

SQL em Oracle Spatial:



SQL em PostGIS:

