UNIVERSIdade FEEVALE

GABRIEL SOARES

ANÁLISE DE DADOS ESPACIAIS NO SQL SERVER 2008

Novo Hamburgo

2012

GABRIEL SOARES

ANÁLISE DE DADOS ESPACIAIS NO SQL SERVER 2008

Trabalho de Conclusão de Curso

apresentado como requisito parcial

à obtenção do grau de Bacharel em

Ciência da Computação pela

Universidade Feevale

Orientador: Me. Edvar Bergmann Araújo

Novo Hamburgo

2012

Agradecimentos

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

Aos amigos e às pessoas que convivem comigo diariamente, minha gratidão, pelo apoio emocional - nos períodos mais difíceis do trabalho.

Resumo

A informação é, de modo geral, um recurso de suma importância em qualquer empresa ou instituição. Ela pode gerar recursos econômicos por reduzir o tempo na tomada de decisões das empresas. Integrar dados convencionais com dados espaciais tornou-se uma necessidade e com isso surgiram os bancos de dados geográficos. Sistemas de Informações Geográficas (SIG) e Sistemas Gerenciadores de Banco de Dados (SGBD) possuem uma área de aplicação muito grande, tanto meio rural, quanto no meio urbano e seu uso facilita muito no momento de tomada de decisões. O presente trabalho realizou uma comparação teórica do módulo de manipulação a dados espaciais dos SGBDs Oracle Spatial, PostGis e SQL Server 2008. Entre os SGBDs destacados, o SQL Server 2008 foi analisado de forma mais detalhada e prática. Nele foram criados exemplos de manipulação de dados espaciais utilizando como estudo de caso a evolução dos municípios dos Vales do Rio do Sinos e Rio Paranhana. Posteriormente a ferramenta ArcGIS 10.1 foi utilizada para demonstrar exemplos semelhantes aos criados no SQL Server 2008, demonstrando assim a possibilidade de se realizar tarefas semelhantes através de uma interface gráfica e não somente com a utilização de códigos T-SQL.

Palavras-chave: Sistemas de Informações Geográficas. SQL Server. Oracle Spatial. PostGis.

Abstract

Information is generally a feature of paramount importance in any company or institution. It can generate economic resources by reducing the time in making business decisions. Integrate data with conventional spatial data has become a necessity and with that came the geographic databases. Geographic Information Systems (GIS) and Management Systems Database (DBMS) have a very large application area, both rural areas as in urban areas and greatly facilitates its use at the time of decision making. This work constitutes a theoretical comparison module handling spatial data DBMSs Oracle Spatial, PostGIS and SQL Server 2008. Among the prominent DBMSs, SQL Server 2008 has been analyzed in more detail and practice. Him were created examples of manipulation of spatial data using as a case study the evolution of the municipalities of the Valley of the Bells River and Rio Paranhana. Subsequently the tool ArcGIS 10.1 was used to demonstrate examples similar to those created in SQL Server 2008, thus demonstrating the possibility of performing similar tasks through a graphical interface and not only with the use of T-SQL code.

Keywords: Geographic Information Systems. SQL Server. Oracle Spatial. PostGis.

**LISTA DE FIGURAS**

Figura 1 – Conjunto de ferramentas do Geoprocessamento\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 13

Figura 2 – Objetos do PostGIS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 23

Figura 3 – Exemplo de consulta em tabela com dados espaciais no PostGIS\_\_\_\_\_\_\_\_25

Figura 4 – Objetos do Oracle Spatial\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 28

Figura 5 – Exemplo de consulta em tabela com dados espaciais no Oracle \_\_\_\_\_\_\_\_\_ 32

Figura 6 – Forma de execução da consulta\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 33

Figura 7 – Objetos do SQL Server 2008\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 35

Figura 8 – Exemplo *geography*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_36

Figura 9 – Exemplo *geometry*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_37

Figura 10 – Decomposição de uma geometria\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 44

Figura 11 – Ferramenta de importação para o SQL Server 2008\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 45

Figura 12 – Divisão atual dos municípios\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 48

Figura 13 – Antiga extensão territorial de São Leopoldo e Taquara\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 49

Figura 14 – Tela inicial do SQL Server 2008\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 51

Figura 15 – Tela do SQL Server 2008 utilizando dados espaciais\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 51

Figura 16 – Inserção dos dados no SGBD\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 52

Figura 17 – Representação de dados codificados e não codificados\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 54

Figura 18 – Inserção de coordenadas decodificadas\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 54

Figura 19 – Inserção de pontos para a criação do antigo município de

Taquara\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 56

Figura 20 – Município de Três Coroas em formato de pontos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 56

Figura 21 – Inserção com a utilização da função *STUnion()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 57

Figura 22 – Inserção através de variável com *STUnion()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 58

Figura 23 – Municípios de São Leopoldo e Taquara nos anos de

1864 e 1908, respectivamente\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 58

Figura 24 – Exibição do SRID\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 59

Figura 25 – Cálculo de área através da função *STArea()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_59

Figura 26 – Utilização das funções *STDimension()* e *STGeometryType()*\_\_\_\_\_\_\_\_\_\_\_ 60

Figura 27 – Utilização de *cursor* para inserir dados na coluna AREA\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 61

Figura 28 – Exibição completa da tabela MUNICIPIOS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 62

Figura 29 – Posição das prefeituras e alguns de seus distritos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 63

Figura 30 – Prefeituras e distritos com um *Buffer* de cinco mil metros de

abrangência\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 64

Figura 31 – Aplicação da função *STDistance()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_65

Figura 32 – Exemplo de utilização da função *STDifference()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_66

Figura 33 – Demonstração de uso da função *STIntersection()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_67

Figura 34 – *Geometry Collection* com ponto, linha e polígono \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 68

Figura 35 – Polígono \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 68

Figura 36 – Demonstração de uso da função *STEquals()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 69

Figura 37 – Utilização da função *STIsClosed()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 70

Figura 38 – Demonstração das funções *STDisjoint()* e *STIntersects()*\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 70

Figura 39 – Identificação dos pontos inicial e final de uma instância \_\_\_\_\_\_\_\_\_\_\_\_\_\_71

Figura 40 – Demais funções do tipo de dados *geography*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_72

Figura 41 – Exportar dados para SHP\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 73

Figura 42 – Tipos de SGBDs suportados pelo ArcGIS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_75

Figura 43 – Configuração de conexão com o SGBD\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 76

Figura 44 – Conexão com o SGBD e armazenamento de arquivos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 76

Figura 45 – Visualização das tabelas da conexão com o SGBD\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 77

Figura 46 – Tela inicial do ArcMap\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 77

Figura 47 – Adicionando uma nova *Layer*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_78

Figura 48 – Opção de calcular a geometria\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_79

Figura 49 – Configuração de cálculo da área ou perímetro\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_80

Figura 50 – Diferenças entre os resultados de área e perímetro\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_80

Figura 51 – Distância entre dois pontos\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 81

Figura 52 – *Buffer* com raio de cinco quilômetros\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 82

Figura 53 – Aplicação da função *Dissolve*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_82

Figura 54 – Resultado da função *Dissolve*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 83

Lista de TABELAS

Tabela 1 – Comparação entre os módulos espaciais\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 49

Lista de QUADROS

Quadro 1 – Comparação entre os módulos espaciais\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 44

Quadro 2 – Descrição da tabela MUNICIPIOS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 53

Quadro 3 – Descrição da tabela MUNICIPIOS\_ANTIGOS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 55

Quadro 4 – Descrição da tabela VERIFICA\_TAMANHO\_MUNICIPIO\_<NOME>\_ 55

Quadro 5 – Legendas das funções *STDimension()* e *STGeometryType()*\_\_\_\_\_\_\_\_\_\_\_\_ 60

Quadro 6 – Descrição da tabela PONTOS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 63

Lista de Abreviaturas e Siglas

|  |  |
| --- | --- |
| API | Application Program Interface |
| DDL | Data Definition Language |
| GB | Gigabyte |
| GML | Geographic Markup Language |
| GPS | Global Position System |
| HTTP | Hypet Text Transfer Protocol |
| IBGE | Instituto Basileiro de Geografia e Estatística |
| ODBC | Open Database Connectivity |
| OGC | Open Geospatial Consortium |
| PL/SQL | Procedural Language/Structured Query Language |
| SFS | Simple Features Service |
| SGBD | Sistema Gerenciador de Banco de Dados |
| SHP | Shapefile |
| SIG | Sistemas de Informações Geográficas |
| SQL | Structured Query Language |
| SRID | Spatial Reference Identifiers |
| T-SQL | Transact Structured Query Language |
| WCS | Web Coverage Service |
| WEB | World Wide Web |
| WFS | Web Feature Service |
| WGS | World Geodetic System |
| WKB | Well Known Binary |
| WKT | Well Konow Text |
| WMS | Web Map Service |
| XML | Extensible Modeling Language |
|  |  |
|  |  |

Sumário

[INTRODUÇÃO 12](#_Toc344476631)

[1 OPENGIS 16](#_Toc344476632)

[1.1 Especificação Abstrata 17](#_Toc344476633)

[1.2 Especificações OGC 19](#_Toc344476634)

[2 BANCOS DE DADOS GEOGRÁFICOS 22](#_Toc344476635)

[2.1 PostGIS 22](#_Toc344476636)

[2.1.1 Representação Geométrica dos Objetos 22](#_Toc344476637)

[2.1.2 Operadores Espaciais 24](#_Toc344476638)

[2.1.3 Indexação Espacial 26](#_Toc344476639)

[2.1.4 Recurso de Carga de Dados 26](#_Toc344476640)

[2.2 Oracle Spatial 27](#_Toc344476641)

[2.2.1 Representação Geométrica dos Objetos 27](#_Toc344476642)

[2.2.2 Operadores Espaciais 30](#_Toc344476643)

[2.2.3 Indexação Espacial 33](#_Toc344476644)

[2.2.4 Recurso de Carga de Dados 34](#_Toc344476645)

[2.3 SQL Server 2008 35](#_Toc344476646)

[2.3.1 Representação Geométrica dos Objetos 35](#_Toc344476647)

[2.3.2 Operadores Espaciais 39](#_Toc344476648)

[2.3.3 Indexação Espacial 43](#_Toc344476649)

[2.3.4 Recurso de Carga de Dados 44](#_Toc344476650)

[2.4 Comparação entre os SGBDs 45](#_Toc344476651)

[3 VALIDAÇÃO DOS RECURSOS ESPACIAIS DO SQL SERVER 2008 47](#_Toc344476652)

[3.1 Estudo de Caso 47](#_Toc344476653)

[3.2 Ferramentas Utilizadas 50](#_Toc344476654)

[3.3 Utilização do Módulo Espacial do SQL Server 2008 50](#_Toc344476655)

[3.3.1 Inserção dos Dados 52](#_Toc344476656)

[3.3.2 Criação e Manipulação das Tabelas 53](#_Toc344476657)

[3.3.3 Funções Espaciais e Análise dos Dados 57](#_Toc344476658)

[3.3.4 Outras Funções Espaciais 66](#_Toc344476659)

[3.4 Exemplos em Ferramenta de Geoprocessamento 73](#_Toc344476660)

[CONCLUSÃO 83](#_Toc344476661)

[REFERÊNCIAS BIBLIOGRÁFICAS 85](#_Toc344476662)

# INTRODUÇÃO

Informação, de um modo geral, é um recurso de suma importância em qualquer empresa ou instituição. Segundo Silva (2010), essa informação pode gerar recurso econômico por diminuir o tempo na tomada de decisão. Assim, essas organizações necessitam cuidar cada vez mais e melhor as suas informações e melhorar o seu banco de dados. Mas para isso é necessária a utilização de sistemas de informação para organizar e manter atualizados os dados que serão utilizados no gerenciamento dessas organizações.

Os dados que organizações e empresas geram são armazenados em bancos de dados, que são ferramentas praticamente indispensáveis nos dias atuais. Quase todas as pessoas no planeta utilizam banco de dados, até mesmo sem perceber, pois as informações neles armazenadas estão em muitos dos lugares que frequentam em suas rotinas diárias, como por exemplo, nos supermercados, bancos, restaurantes, farmácias, postos de gasolina, escolas, dentre outros.

A necessidade em integrar dados convencionais com dados espaciais, fez com que surgissem os bancos de dados geográficos. Alguns Sistemas Gerenciadores de Banco de Dados (SGBD), como por exemplo, o Oracle, PostgreSQL e o SQL Server 2008, possuem módulos específicos que permitem a manipulação de dados espaciais. Através da utilização destes módulos espaciais é possível realizar a integração entre os dados convencionais e espaciais, o que é fundamental, pois permite a realização de uma análise de diferentes tipos de dados e onde estes dados ocorrem no espaço.

Segundo Silva (2010), o geoprocessamento pode ser definido como um conjunto de tecnologias voltadas à coleta e ao tratamento de informações espaciais para um objetivo específico, conforme demonstra a figura 1. Essas atividades são executadas por sistemas chamados de Sistemas de Informação Geográfica (SIG). Eles são destinados ao processamento de dados georreferenciados desde a sua coleta até a geração de produtos como mapas, relatórios e arquivos digitais, oferecendo recursos para armazenamento, gerenciamento, manipulação e análise dos dados.



Figura 1 – Conjunto de técnicas do geoprocessamento.

Fonte: Silva, 2010.

Devido ao aumento na utilização de SIGs, tem se buscado cada vez mais uma solução para o gerenciamento dos dados (espaciais, alfanuméricos, ou imagens) manipulados por esses sistemas. (CÂMARA, 2005). Para Silva (2010), SIG são ferramentas que permitem armazenar, analisar, recuperar e manipular grandes quantidades de dados espaciais. Os SIGs são técnicas de manipulação de dados variáveis espacialmente, podendo ou não associar-se a dados descritivos.

Conforme Silva (2010), o termo espacial refere-se a qualquer tipo de informação sobre localização e pode incluir informações de engenharia, cartografia, sensoriamento remoto, etc. O termo geográfico refere-se somente à localização da informação sobre a superfície da Terra ou próxima dela em escalas do mundo real e no espaço do mundo real.

O termo SIG é aplicado para sistemas que realizam o tratamento computacional de dados geográficos. A principal diferença de um SIG para um sistema de informação convencional é sua capacidade de armazenar tanto os atributos descritivos como as geometrias dos diferentes tipos de dados geográficos (CÂMARA, 2005).

A aplicabilidade de SIGs é vasta, podendo ser utilizada com grande propriedade na área rural, urbana e ambiental, pois pode auxiliar na tomada de decisões. Como exemplos de aplicações pode-se citar o controle de tráfego urbano de uma cidade, o registro e manutenção de dados a respeito de lotes urbanos e a criação de mapas interativos em páginas web (SANTOS, 2006).

Outro exemplo de aplicação de dados espaciais pode ser uma pesquisa policial sobre o índice de criminalidade de uma determinada cidade ou região, através de, por exemplo, filtros por perímetros pré-definidos. Pode-se ainda usar os dados espaciais para calcular a área e a porcentagem de floresta virgem já desmatada.

Dados geográficos são aqueles que possuem uma dimensão espacial, ou uma localização, diretamente ligada ao mundo geográfico real como as imagens de satélites de sensoriamento remoto, os dados de inventários cadastrais, os dados ambientais coletados em campo e os modelos numéricos de terreno (VINHAS, 2006).

Bancos de dados geográficos possuem como necessidade básica uma topologia específica. Silva (2010) descreve que topologia define a relação posicional de características ao definir suas propriedades. Inclui informações sobre que rótulos estão ligados a cada característica, como os pontos que estão ligados uns aos outros e que os polígonos são formados pela ligação de linhas. Qualquer fenômeno topológico pode ser reduzido a um dos três conceitos topológicos básicos: ponto, linha e polígono.

Um plano de informação é formado por um conjunto de geometrias que possuem um mesmo conjunto de atributos. As geometrias por sua vez, podem ser formadas por um único elemento ou por um conjunto de elementos. E os elementos são formados por tipos geométricos primitivos como ponto, cadeia de linhas e polígonos (QUEIROZ, 2005).

O método de pesquisa utilizado no presente trabalho, para fins de investigação, é o método experimental. Segundo Prodanov (2009), este método consiste em submeter os objetos de estudo à influência de certas variáveis, em condições controladas e conhecidas pelo investigador, para observar os resultados que a variável produz no objeto.

A partir desta afirmação de aplicação do método experimental, foi escolhido, como estudo de caso, a evolução de alguns municípios dos vales do Rio do Sinos e Rio Paranhana. Esta escolha se deu pelo fato do pesquisador e também a Universidade Feevale estarem localizados geograficamente nesta região e consequentemente o presente trabalho ser mais bem compreendido por todos que, em sua maioria, aqui residem.

O presente trabalho teve sua elaboração devido ao fato de que dados espaciais são muito interessantes e muitas vezes desconhecidos por grande parte das pessoas que em sua maioria apenas ouviram falar em GPS para uso veicular e mesmo assim não conhecem o seu funcionamento.

Além de o tema sobre dados espaciais ser muito interessante, o estágio em que se encontra a sua teoria é insuficiente, pois grandes dificuldades surgiram no decorrer da primeira etapa (pesquisa bibliográfica) devido a não existirem muitas bibliografias disponíveis e as existentes, estando em sua grande parte em outra língua. Na segunda etapa (prática), as dificuldades não foram menores, pois em várias situações era sabida a existência de alguma função ou procedimento espacial no SQL Server 2008 e a sua elaboração muitas vezes era complexa devido ao fato de não haver exemplos ou demonstrações nem mesmo em páginas *web* da Microsoft.

No capítulo um, será demonstrado o que é o consórcio OpenGIS, qual a finalidade da sua criação e quais são as suas especificações para que exista a interoperabilidade entre diferentes ferramentas. O capítulo dois, explicará como funciona a representação geométrica dos objetos, operadores espaciais, indexação espacial e os recursos de carga de dados dos módulos espaciais dos SGBDs PostgreSQL, Oracle e SQL Server 2008.

E no capítulo três será realizada a validação dos recursos espaciais do SQL Server 2008. Isto ocorrerá através de um estudo de caso sobre a evolução histórica dos municípios dos vales do Sinos e Paranhana. Por fim, outros exemplos serão exibidos através da ferramenta SIG ArcGIS, com o intuito de deixar clara a diferença e dificuldade entre a manipulação de dados através de códigos (SQL Server 2008) ou através de interface gráfica (ArcGIS).

# OPENGIS

O mercado mundial de geoprocessamento se desenvolveu sem a preocupação de integração de informações geográficas entre sistemas. Formatos e estruturas de dados incompatíveis têm dificultado a integração de informações geográficas e limitado o uso da tecnologia. A indústria de Geoprocessamento, pressionada pela demanda da comunidade usuária, resolveu unir-se para avaliar a questão e projetar o futuro da tecnologia (SANTOS, 2006).

A popularização das geotecnologias fez com que o volume de fontes independentes de dados geográficos aumentasse. Essa realidade disponibiliza uma grande oportunidade para o intercâmbio de dados, reduzindo custos e agilizando processos. Entretanto, para que isso ocorra, os aplicativos devem ser capazes de processar dados oriundos de outras fontes e de si mesmos (BORGES, 2006).

OpenGIS ou *Open Geospatial Consortium* (OGC), é um consórcio internacional de padrões composto de 451 empresas, agências governamentais e universidades que participam em um processo de consenso para o desenvolvimento de padrões de interface disponíveis publicamente. OGC e OpenGIS são marcas registradas do *Open Geospatial Consortium* (OGC, 2012).

Criado em 1994, o OGC possui como objetivo criar padrões abertos para garantir a interoperabilidade entre sistemas que gerenciam informações geo-espaciais. Inicialmente o consórcio era chamado de OpenGIS. Quando passou a ser chamado de OGC, suas especificações e documentos criados receberam a denominação de OpenGIS. Produtos que implementam ou atendem às especificações da OGC também estão associados à designação OpenGIS (LOURENÇO, 2008).

Entende-se por interoperabilidade a capacidade de compartilhar e trocar informações e processos entre ambientes computacionais heterogêneos, autônomos e distribuídos (YUAN, 1998).

## Especificação Abstrata

O comitê Técnico do OGC tem desenvolvido uma arquitetura para o auxílio a tecnologias geoespaciais e interoperabilidade de dados que se chama Especificação Abstrata do OpenGIS. Esta especificação fornece a base conceitual para a maioria das atividades de desenvolvimento de especificações do OGC (OGC, 2012).

Interfaces abertas e protocolos são construídos e referenciados com a especificação abstrata, permitindo assim a interoperabilidade entre os mais diversos tipos de sistemas de processamento espacial existentes.

Quando um novo software é desenvolvido, os membros do OGC analisam os requisitos de especificação da implementação e quando chegam a um consenso positivo, este software passa a contar com o certificado OGC (THOMÉ, 1998).

A especificação abstrata disponibiliza um modelo de referência para o desenvolvimento de especificações de implementação conforme os padrões do OGC. Este modelo é chamado de Modelo Abstrato e possui a descrição de como o software de SIG funcionará.

O Modelo Abstrato está dividido em vinte tópicos, incluso o tópico zero que representa o relacionamento entre os demais. Para cada um dos tópicos existe um subtópico, onde é descrito o Modelo Essencial, que representa os fatos do mundo real. A seguir a descrição dos tópicos (OGC, 2012):

* ***Overview* (Visão Geral):** introdução e roteiro para a especificação abstrata;
* ***Feature Geometry* (Geometria de Traços):** representação abstrata dos fenômenos do mundo real, associada a uma localização relativa à Terra;
* ***Spatial Referencing by Coordinates* (Referenciamento Espacial Através de Coordenadas):** associa localizações no espaço a tuplas de coordenadas geométricas em um espaço matemático e associa estes valores no mundo real;
* ***Locational Geometry Structures* (Estruturas Geometria Locacionais):** fornece modelos essenciais e abstratos para que a tecnologia SIG seja amplamente utilizada;
* ***Stored Functions and Interpolation* (Funções de Interpolação e Armazenados):** fornece modelos essenciais e abstratos para que a tecnologia seja amplamente utilizada em toda a paisagem do SIG. São definidas uma única vez e utilizadas em todos os pontos de um domínio;
* ***Features* (Recursos):** característica geográfica associada a uma localização relativa à Terra;
* ***Schema for Coverage Geometry and Functions* (Esquema da Geometria de Cobertura e Funções):** esta norma define um esquema conceitual para as características espaciais de coberturas;
* ***Earth Imagery* (Imagens da Terra):** exibe modelos essenciais e abstratos para que a tecnologia seja amplamente utilizada através de um SIG;
* ***Relationships Between Features* (Relações entre as Características):** exibe uma abstração para as relações entre as entidades do mundo real;
* ***Feature Collections* (Coleções de Recursos):** é um objeto abstrato composto de instâncias de recursos;
* ***Metadata* (Metadados):** Associação de metadados;
* ***The OpenGis Service Architecture* (Serviço de Arquitetura OpenGis):** fornece a definição das partes essenciais de uma visão computacional de um sistema de informação para processamento geoespacial;
* ***Catalog Services* (Serviços de Catálogo):** abrange os serviços de acesso às informações geoespaciais;
* ***Semantics and Information Communities* (Semântica e Comunidades de Informação):** permite que grupos de profissionais gerenciem com eficiência a semântica de suas próprias coleções de dados geográficos, criando semânticas distintas;
* ***Image Exploitation Services* (Serviços de Exploração de Imagem):** descreve as categorias dos serviços de exploração de imagem necessárias;
* ***Image Coordinate Transformation Services* (Serviços de Transformação de Coordenadas de Imagens):** abrange os serviços de conversão de coordenadas da imagem;
* ***Location Based Mobile Services* (Localização Baseada em Serviços Móveis):** nunca formalmente aprovada;
* ***Geospatial Digital Rights Management Reference Model* (GeoDRM RM):** modelo de referência para o gerenciamento dos direitos e funcionalidades para recursos geoespaciais;
* ***Geographic Information* - *Linear Referencing* (Informação Geográfica - Referência Linear):** define uma descrição dos dados e as operações necessárias para a utilização e apoio ao referenciamento linear;
* ***Observations and Measurements* (Observações e Medições):** define um esquema conceitual para observar os recursos envolvidos na amostragem.

## Especificações OGC

Dentre as várias especificações já definidas pelo OGC, as mais utilizadas são (LOURENÇO, 2008):

* ***Web Map Service* (WMS):** fornece um protocolo HTTP simples para solicitar imagens de mapas geo-registrados de uma ou mais bases de dados geoespaciais distribuídas. Uma requisição define as camadas de informações geográficas e as áreas de interesse para serem processadas. A resposta da requisição é o recebimento de uma ou mais imagens que correspondem a sua visualização do mapa, de acordo com as camadas (vetoriais ou matriciais) solicitadas, que podem ser exibidas em um navegador;
* ***Web Feature Service* (WFS):** *Web Service que* é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Vem sendo cada vez mais empregado no desenvolvimento de aplicações Web, pois possibilita a interação entre aplicações desenvolvidas em diferentes plataformas. Com o uso desta tecnologia, as aplicações podem enviar e receber dados em formato XML;
* ***Web Coverage Service* (WCS):** permite o acesso interoperável a dados geoespaciais como imagens de satélite, fotos aéreas digitais, dados de elevação digital e outros fenômenos representados por valores em cada ponto de medição;
* ***Simple Feature Specifications* (SFS):** essa especificação se baseia no padrão SQL (*Strutured Query Language*) e define um esquema padrão para o armazenamento, leitura, análise e atualização de dados geográficos através da ODBC API (*Application Programming Interface*);
* ***Geography Markup Language* (GML):** é um padrão baseado em XML para expressar características geográficas. Desenvolvido para permitir o transporte e armazenamento de informações geográficas. Serve tanto como uma linguagem de modelagem para sistemas geográficos, quanto como um formato de intercâmbio aberto para transações geográficas na internet. Um documento GML é descrito utilizando um esquema GML. É possível que usuários e desenvolvedores descrevam conjuntos de dados geográficos genéricos que contenham pontos, linhas e polígonos.

As especificações do OGC são muito importantes, pois realizam a integração e comunicação entre diferentes sistemas e permitem que a interoperabilidade exista quando se acessa (ou realiza intercâmbio) dados geoespaciais. Além disso, definem um esquema padrão para a utilização de dados geográficos através da ODBC API.

# BANCOS DE DADOS GEOGRÁFICOS

Segundo Lourenço (2008), SGBDs são sistemas que oferecem serviços de armazenamento, manutenção e consulta de banco de dados. As principais características de um SGBD são usabilidade, integridade, manutenibilidade, confiabilidade, segurança e desempenho.

Os dados espaciais descrevem a posição, forma e orientação dos objetos no espaço. Estes objetos podem ser locais sinalizados com um simples ponto ou objetos mais complexos como países, estradas e lagos. (MICROSOFT, 2008)

## PostGIS

Segundo Queiroz (2005), o PostGIS é a extensão espacial do SGBD PostgreSQL que é gratuito e de código fonte aberto. Desenvolvida pela *Refractions Research*, esta extensão, adiciona entidades geográficas ao PostgreSQL, seguindo a especificação SFS do OGC.

### Representação Geométrica dos Objetos

Conforme Paiva (2007), caracteriza-se que ponto, linha, polígono, multiponto, multilinhas, multipolígonos, e coleção de geometrias, ilustrados na figura 2, são os tipos de objetos suportados pelo OpenGis. Esta extensão possui ainda, para atender as especificações da OGC, duas tabelas de metadados, onde uma armazena a descrição das feições dos dados espaciais, denominada *GEOMETRY\_COLUMNS*, e a outra tabela, carrega os identificadores numéricos e descrições textuais de sistemas de coordenadas utilizadas no banco de dados espacial, sendo denominada de *SPATIAL\_REF\_SYS*.

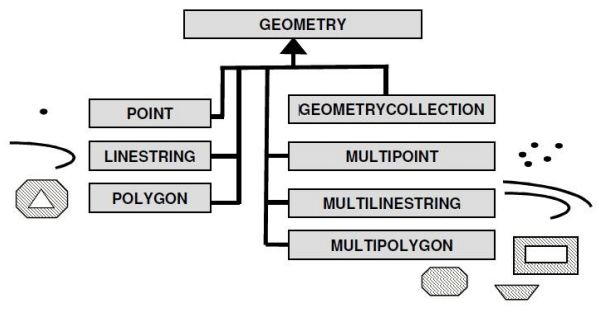


Figura 2 – Objetos do PostGIS.

Fonte: PostGIS, 2011.

Estes tipos de dados são definidos da seguinte forma:

* ***Point*:** define um único local (ponto) com duas dimensões;
* ***LineString*:** define uma única linha;
* ***Polygon*:** reunião de diversos pontos pertencentes a uma mesma dimensão;
* ***GeometryCollection*:** é uma coleção de zero ou mais geometrias;
* ***MultiPoint*:** é uma coleção de zero ou mais pontos. O limite de uma instância *MultiPoint* é vazio;
* ***MultiLineString*:** é uma coleção de zero ou mais linhas;
* ***MultiPolygon*:** é uma coleção de zero ou mais polígonos.

Segundo Lourenço (2008), a criação de uma tabela com dado espacial é realizada em duas etapas. Na primeira, define-se os atributos alfanuméricos e na segunda, insere-se a coluna espacial utilizando a função *AddGeometryColumn*. Essa função é responsável pela manutenção da tabela de metadados *GEOMETRY\_COLUMNS*. Esta tabela recebe como parâmetros os seguintes atributos:

* Nome do banco de dados;
* Nome da tabela que contém a coluna espacial;
* Nome da coluna espacial;
* Sistema de coordenadas;
* Tipo de coluna espacial;
* Dimensão em que se encontram as coordenadas dos dados.

### Operadores Espaciais

O PostGIS possui diversas funções para as mais variadas situações, dentre as quais, algumas serão listadas a seguir, conforme PostGIS (2011):

* ***Distance*:** retorna a distância mínima (em metros), em linha reta entre duas instâncias;
* ***Equals*:** retorna verdadeiro se as duas instâncias forem iguais. Caso contrário, retornará falso;
* ***Disjoint*:** retorna verdadeiro se as geometrias não se cruzam espacialmente. Caso contrário, retornará falso;
* ***Intersects*:** retorna verdadeiro se as geometrias se cruzam espacialmente. Caso contrário, retornará falso;
* ***Touches*:** retorna verdadeiro se as instâncias possuírem pelo menos um ponto em comum;
* ***Crosses*:** retorna verdadeiro se as instância fornecidas possuírem alguns pontos interiores em comum;
* ***Within*:** retorna verdadeiro se a instância “A” estiver completamente dentro da instância “B”;
* ***Area*:** retorna a área de uma determinada instância em metros quadrados;
* ***Difference*:** retorna uma instância “C” de uma instância “A” que não se cruza com uma instância “B”;
* ***Centroid*:** retorna o centro de uma instância;
* ***Union*:** retorna o objeto que representa a união de duas instâncias;
* ***Buffer*:** retorna as instâncias existentes dentro da área de abrangência definida.

Na figura 3 é exemplificada uma consulta em uma tabela espacial através da utilização do operador *buffer* na ferramenta pgAdmin, que é utilizada para realizar a manipulação dos dados no SGBD PostgreSQL.

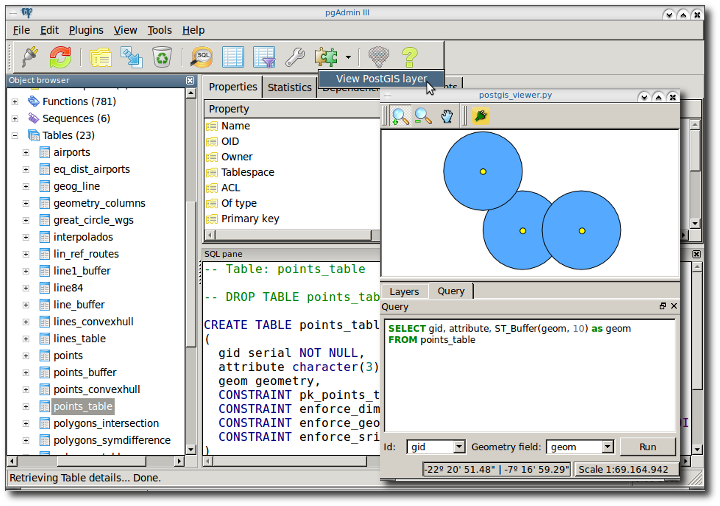


Figura 3 – Exemplo de consulta em tabela com dados espaciais no PostGIS.

Fonte: Carrillo, 2012.

### Indexação Espacial

Segundo PostGIS (2011), o PostGIS suporta três tipos de índice:

* ***R-Tree*:** são utilizados para indexar dados de SIG;
* ***B-Tree:*** são utilizados para dados que podem ser ordenados ao longo de um eixo;
* ***GiST*:** é utilizado um índice *R-Tree* implementado sobre o índice *GiST* para indexar dados de SIG.

Conforme Queiroz (2005), os índices espaciais são utilizados em consultas que envolvem predicados espaciais como as consultas por janelas, onde um retângulo envolvente é informado e as geometrias que interagem com ele devem ser recuperadas rapidamente.

### Recurso de Carga de Dados

Após a criação de uma tabela espacial no PostGIS, o usuário já pode realizar a importação de dados GIS para o banco de dados. A importação dos dados pode ser realizada de duas formas (POSTGIS, 2011):

* Converter os dados para texto e então, utilizando comandos SQL, importar os mesmos via script;
* Utilizar o aplicativo Shp2pgsql, que converte os arquivos SHP (*Shapefile*) e os insere no PostGIS.

Arquivos SHP são arquivos que não possuem sobrecarga de processamento. Em relação a outras fontes de dados, possuem como vantagem uma maior velocidade de desempenho e maior facilidade na edição dos dados (ESRI, 1998).

## Oracle Spatial

O Oracle Spatial, segundo Murray (2003), é uma extensão espacial desenvolvida sobre o modelo objeto-relacional do SGBD Oracle. Esta extensão é baseada nas especificações do OGC e contém um conjunto de funcionalidades e procedimentos que permitem manipular dados espaciais de representação vetorial em um banco de dados Oracle.

Inclui suporte integral a 3D e serviços web para gerenciar todos os dados geoespaciais incluindo dados de vetores e de varredura, topologia e modelos de rede. Ele foi criado para atender às necessidades das aplicações avançadas de SIG, como gerenciamento de terras, serviços públicos e defesa/segurança interna. O formato espacial nativo e aberto do Oracle elimina o custo de sistemas proprietários separados e tem suporte de todos os principais fornecedores de GIS (ORACLE, 2003).

Esta extensão espacial fornece um esquema SQL e funções que facilitam o armazenamento, recuperação, atualização e consulta de coleções de características espaciais em um banco de dados Oracle. Esta extensão consiste nos seguintes componentes (ORACLE,2003):

* Um esquema que define o armazenamento, sintaxe e semântica dos tipos de dados geométricos suportados;
* Um mecanismo de indexação espacial;
* Um conjunto de operadores e funções para a realização de consultas na área de interesse, junção de consultas espaciais e outras operações de análise espacial;
* Aplicativos administrativos.

### Representação Geométrica dos Objetos

Segundo Queiroz (2005), o modelo de dados do Oracle Spatial consiste em uma arquitetura hierárquica de elementos, geometrias e planos de informação (*layers*). Uma geometria é uma sequência ordenada de vértices que são conectados por segmentos de reta ou arcos circulares. A semântica da geometria é determinada pelo seu tipo. Oracle Spatial suporta vários tipos primitivos e geometrias compostas por coleções desses tipos, incluindo de duas dimensões (ORACLE, 2003). A figura 4 demonstra os tipos de objetos aceitos pelo Oracle Spatial.

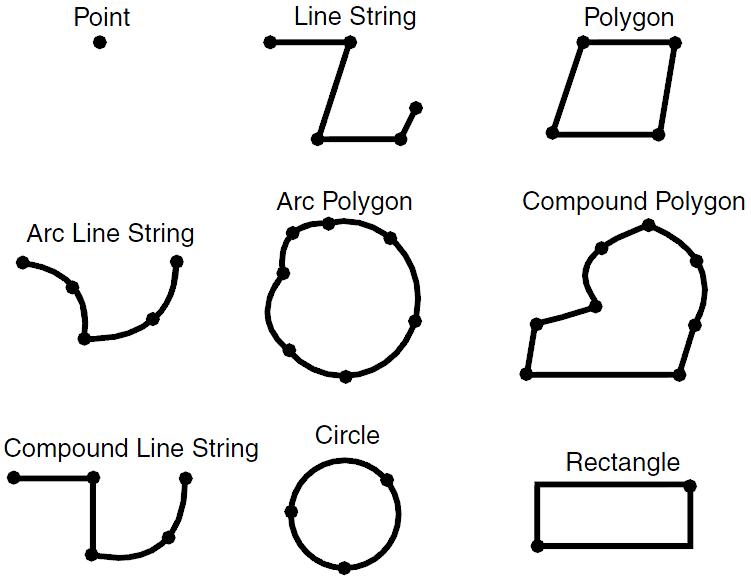


Figura 4 – Objetos do Oracle Spatial.

Fonte: Oracle, 2003.

Pontos bidimensionais são elementos constituídos de duas coordenadas, X e Y, que muitas vezes correspondem a longitude e latitude, como por exemplo, sequências de linhas que são compostas de um ou mais pares de pontos que definem o segmento da linha e polígonos que são formados pela conexão de uma sequência de linhas que formam um anel fechado. Um exemplo prático pode ser um ponto que representa onde está localizada uma determinada loja, uma linha que demonstra a localização de uma estrada e um polígono que pode vir a representar um país, estado, cidade ou região.

Em uma tabela espacial, a geometria de um objeto fica armazenada em uma coluna de nome *SDO\_GEOMETRY*. Dessa forma, cada instância da geometria é armazenada em uma linha, e o conjunto de todas as geometrias de uma mesma tabela formam um plano (PAIVA, 2007).

*SDO\_GEOMETRY* é o nome dado ao tipo de objeto que o ORACLE SPATIAL utiliza para representar os dados espaciais. Nesse objeto é que ficam armazenadas informações sobre a geometria em si como suas coordenadas, seu tipo e projeção. Um objeto *SDO\_GEOMETRY* é definido como mostrado a seguir (LOURENÇO, 2008):

CREATE TYPE sdo\_geometryASOBJECT(

SDO\_GTYPE NUMBER,

SDO\_SRID NUMBER,

SDO\_POINT SDO\_POINT\_TYPE,

SDO\_ELEM\_INFO SDO\_ELEM\_INFO\_ARRAY,

SDO\_ORDINATES SDO\_ORDINATE\_ARRAY);

O tipo de objeto *SDO\_GEOMETRY* possui atributos que são especificados a seguir (ORACLE, 2003):

* ***SDO\_GTYPE*:** este atributo indica o tipo da geometria. É formado por quatro números, aonde os dois primeiros indicam a dimensão da geometria e os dois últimos indicam o tipo. Os tipos podem ser:
  + 00 – Tipo não conhecido;
  + 01 – Ponto;
  + 02 – Linha ou curva;
  + 03 – Polígono;
  + 04 – Coleção;
  + 05 – Multipontos;
  + 06 – Multilinhas ou multicurvas;
  + 07 – Multipolígonos.
* ***SDO\_SRID*:** atributo que pode ser utilizado para identificar um sistema de coordenadas (sistema de referência espacial) que serão associadas a geometria. Se SDO\_SRID for nulo, nenhum sistema de coordenada estará associado a geometria;
* ***SDO\_POINT*:** este atributo somente será informado se o tipo da geometria for do tipo ponto*. SDO\_POINT* é utilizado para representar as coordenadas do ponto (X, Y, Z);
* ***SDO\_ELEM\_INFO*:** este atributo é definido usando um vetor de números de tamanho variável, que armazena as características dos elementos que compõem a geometria. Coordenadas de cada um dos elementos são armazenadas em um vetor de nome *SDO\_ORDINATES*:
  + ***SDO\_STARTING\_OFFSET*:** indica a posição da primeira coordenada do elemento no *SDO\_ORDINATES*;
  + ***SDO\_ETYPE*:** indica o tipo do elemento;
  + ***SDO\_INTERPRETATION*:** indica como deverá ser feita a interpretação do elemento junto com o *SDO\_ETYPE*.
* ***SDO\_ORDINATES*:** é um vetor de tamanho variável responsável por armazenar as coordenadas da geometria.

### Operadores Espaciais

O Oracle Spatial dispõe de um conjunto de operadores e funções espaciais para dar suporte as consultas e operações espaciais. Os operadores são utilizados na cláusula *WHERE* e necessitam de indexação espacial, ou seja, somente podem ser utilizados em colunas espaciais já indexadas.

As funções são definidas como subprogramas em PL/SQL, e são utilizadas tanto na cláusula *WHERE* quanto em sub-consultas, não sendo necessária a indexação, podendo ser aplicadas em colunas espaciais não indexadas (QUEIROZ, 2006).

Os principais operadores disponíveis no Oracle Spatial:

* ***SDO\_FILTER*:** especifica quais geometrias podem interagir com determinada geometria;
* ***SDO\_JOIN*:** realiza uma junção espacial com base em um ou mais relacionamentos topológicos;
* ***SDO\_NN*:** determina quais são as geometrias vizinhas de um determinada geometria;
* ***SDO\_NN\_DISTANCE*:** retorna a distância do objeto retornado pelo operador *SDO\_NN*;
* ***SDO\_RELATE*:** determina se duas geometrias se interagem;
* ***SDO\_WITHIN\_DISTANCE*:** determina se duas geometrias estão dentro de uma distância especificada;
* ***SDO\_CONTAINS*:** recebe duas geometrias como parâmetro e retorna verdadeiro se o interior e a borda de uma geometria estiverem completamente dentro do interior da outra geometria;
* ***SDO\_COVERS*:** este método é o oposto de *SDO\_COVERS*, pois ele retorna verdadeiro se uma geometria é completamente coberta por outra e ainda suas bordas se interceptam;
* ***SDO\_COVEREDBY*:** recebe duas geometrias como parâmetro e retorna verdadeiro se o interior de uma geometria está completamente dentro da outra geometria e suas bordas se interceptam;
* ***SDO\_EQUAL*:** recebe duas geometrias como parâmetro e retorna verdadeiro se as geometrias possuírem a mesma borda e o mesmo interior;
* ***SDO\_INSIDE*:** este método é oposto de *SDO\_CONTAINS*, pois ele retorna verdadeiro se uma geometria contém a outra por completo;
* ***SDO\_ON*:** esse método recebe duas geometrias como parâmetro e retorna verdadeiro se o interior e a borda de uma geometria estão na borda da outra geometria;
* ***SDO\_OVERLAPBDYDISJOINT*:** recebe como parâmetro uma geometria do tipo polígono e outra do tipo linha. Retorna verdadeiro se a origem da linha for fora do polígono, mas se a mesma terminar dentro do polígono, retorna falso;
* ***SDO\_OVERLAPBDYINTERSECT*:** recebe duas geometrias como parâmetro e retorna verdadeiro se houver interseção da borda com o interior;
* ***SDO\_TOUCH***: recebe como parâmetro duas geometrias. Retorna verdadeiro se apenas as bordas das geometrias se interceptarem, caso contrário, retorna falso.

Abaixo, na figura 5, é ilustrada uma consulta sendo realizada em uma tabela contendo dados espaciais através da ferramenta SQL Developer, que é utilizada para a manipulação dos dados no SGBD Oracle (PETEK, 2006).

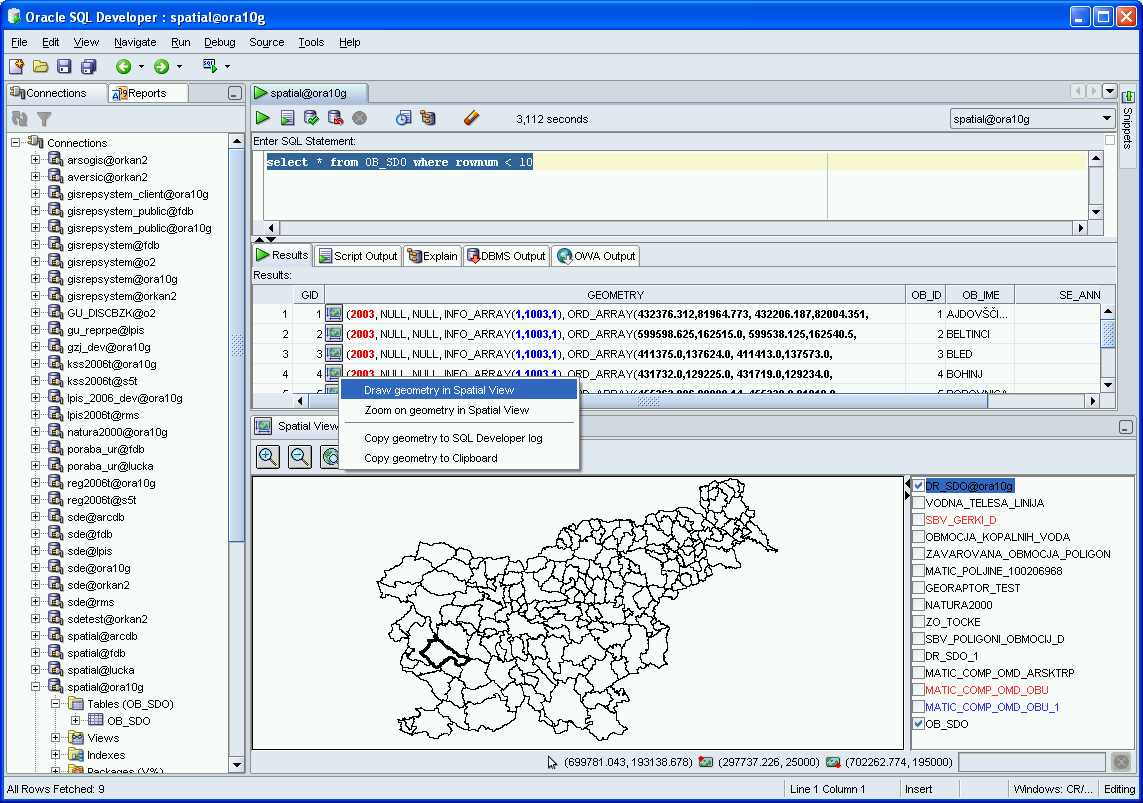


Figura 5 – Exemplo de consulta em tabela com dados espaciais no Oracle.

Fonte: Petek, 2006.

O modelo de consulta do Oracle Spatial é baseado em duas etapas denominadas de primeiro e segundo filtro. A operação do primeiro filtro sobre um grande conjunto de dados de entrada produz um conjunto candidato menor, que contém um conjunto mínimo de resultados e ainda pode conter mais resultados (ORACLE, 2003).

É utilizado um índice espacial para implementar o primeiro filtro. Não é obrigatório o uso do primeiro e segundo filtro. Em alguns casos, usando apenas o primeiro filtro é suficiente.

Segundo Queiroz (2005), o primeiro filtro considera a aproximação das geometrias utilizando o critério do Mínimo Retângulo Envolvente (*Minimum Bounding Rectangle* -MBR). Esse filtro possui um baixo custo e ao ser processado, seleciona um subconjunto menor de geometrias candidatas, na qual será aplicado o segundo filtro. Já o segundo filtro, trabalha com as geometrias exatas, sendo, portanto, mais caro que o primeiro. Ao ser processado o segundo filtro, já se obtém o resultado esperado na consulta. A figura 6 demonstra como o Oracle Spatial executa suas consultas.

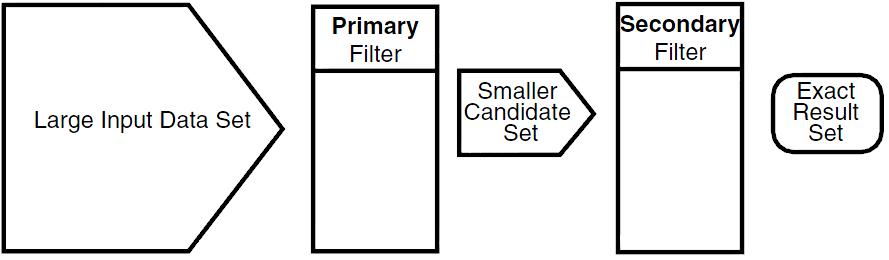


Figura 6 – Forma de execução da consulta.

Fonte: Oracle, 2003.

### Indexação Espacial

Um índice espacial, como qualquer outro índice, fornece um mecanismo para agilizar as pesquisas. Mas neste caso, o mecanismo é baseado em critérios espaciais tais como intersecção e contenção. Todo índice espacial é considerado um índice lógico. É necessário um índice espacial para (ORACLE, 2003):

* Encontrar objetos de dados espaciais indexados que interajam com um determinado ponto ou área de interesse;
* Encontrar os pares de objetos de dois dados espaciais que interajam espacialmente uns com os outros.

O Oracle Spatial permite que o usuário utilize dois índices espaciais, o *R-tree* (padrão) e o *Quadtree*, ou ainda os dois. No entanto, utilizar o índice *Quadtree* não é aconselhável pela Oracle, que aconselha que os usuários utilizem o índice *R-tree*. Com o desenvolvimento de novas versões do Oracle Database, foram adicionadas algumas melhorias para o índice *R-tree* e por esta razão, seu desempenho é superior ao do índice *Quadtree* que se tornou uma indexação obsoleta.

Para se criar um índice *R-tree*, o usuário pode utilizar os parâmetros padrões ou pode especificar parâmetros como o tamanho da memória utilizada e o número de dimensões a serem indexadas. A ação de inserir, remover ou modificar geometrias de uma determinada tabela pode degradar o desempenho do índice espacial. Para contornar este problema, essa extensão dispõe de um conjunto de funções responsáveis por avaliar o desempenho dos índices e para reconstruí-los (LOURENÇO, 2008).

### Recurso de Carga de Dados

Conforme Lourenço (2008), para realizar carga de dados, o Oracle Spatial possui uma ferramenta chamada Oracle Loader ou SQL \* Loader que possui como finalidade mover dados de outra origem para dentro de uma base de dados Oracle através de uma tabela que possua uma coluna do tipo *SDO\_GEOMETRY*. O Oracle Spatial utiliza tabelas que podem ser acessadas e carregadas através da sintaxe padrão do SQL.

Segundo a Oracle (2003), o processo de carga de dados pode ser classificado em duas categorias que seguem:

* **Carregamento em massa dos dados:** este processo é utilizado para carregar grandes volumes de dados para dentro do banco de dados e utiliza o SQL \* Loader como ferramenta para a realização do carregamento dos mesmos;
* **Operações de inserção transacional:** este processo é utilizado para inserir quantidades relativamente pequenas de dados no banco de dados, utilizando a instrução de *INSERT* do SQL.

Para a importação de dados no formato SHP, a Oracle disponibiliza a ferramenta shp2sdo. Esta ferramenta é responsável por converter os arquivos SHP em um formato reconhecido pela ferramenta SQL \* Loader (LOURENÇO, 2008).

## SQL Server 2008

A partir de sua versão 2008, o SQL Server começou a oferecer aos seus usuários um novo módulo, de manipulação de dados espaciais, que vem junto com o SGBD tradicional. Neste módulo espacial, dois novos tipos de dados foram adicionados, o *geometry,* onde os dados são tratados como se estivessem em um plano e o *geography*, onde os dados são tratados levando em conta a curvatura da Terra.

### Representação Geométrica dos Objetos

O SQL Server oferece suporte a dois tipos de dados espaciais: *geometry* (geometria) e *geography* (geografia). Estes tipos de dados oferecem suporte a onze objetos de dados espaciais ou tipos de instâncias. No entanto, apenas sete tipos de instâncias podem ser instanciados. Estas instâncias estão marcadas em azul na figura 7. (MICROSOFT, 2008)

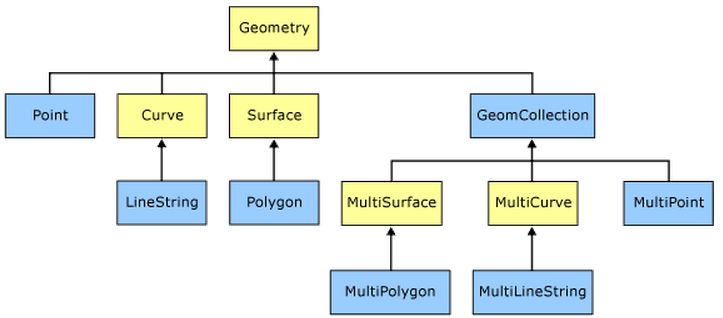


Figura 7 – Objetos do SQL Server 2008.

Fonte: Microsoft, 2008.

O tipo de dados *geography* possui como característica mais importante a forma de armazenar seus dados geodesicamente, levando em conta a curvatura da Terra. Quando é executada uma operação utilizando o tipo de dados *geography*, o SQL Server utiliza dados angulares para chegar ao resultado. A realização destes cálculos é baseada no modelo elipsoide da Terra definido pelo sistema de referência espacial dos dados em questão (AITCHISON, 2009).

Por exemplo, se o usuário quiser traçar uma linha conectada por dois pontos sobre a superfície da Terra utilizando o tipo de dados *geography*, a linha seguiria a superfície da referência elipsoidal. Cada linha desenhada entre dois pontos é, na realidade, um grande arco elíptico. A linha na superfície da Terra é formada por um plano contendo dois pontos e uma referência elipsoidal no centro, conforme ilustra a figura 8 (AITCHISON, 2009).

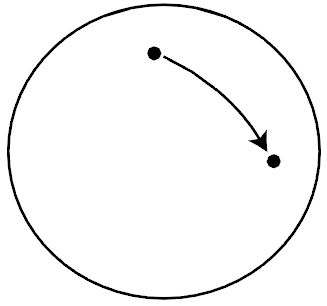


Figura 8 – Exemplo *geography*.

Fonte: Aitchison, 2009.

Dados do tipo *geography* são baseados em três dimensões, modelo de mundo redondo, e então se deve utilizar o sistema geográfico de coordenadas para especificar as posições de cada ponto criado.

Conforme Aitchison(2009), desde quando foi desenvolvido, o tipo de dados *geography* define a coordenada de seus pontos utilizando os sistemas de latitude e longitude, que são definidos em graus. Segundo Lourenço (2008), os comprimentos e áreas são normalmente medidos em metros ou metros quadrados. No entanto, podem depender do SRID (*Spatial Reference Identifiers*) do objeto.

Cada vez que o usuário armazena um item de dados usando o tipo *geography*, ele deverá fornecer o SRID apropriado, correspondente ao sistema de referencia do qual as coordenadas foram obtidas. O SRID fornecido deve, portanto, estar relacionado na lista de referências espaciais suportadas da tabela sys.spatial\_reference\_systems.

Ao utilizar este tipo de dados é necessário especificar a orientação e o local preciso do objeto. Caso contrário, ele não terá significado ou será ambíguo. Por exemplo, será que um anel ao redor do equador descreve o hemisfério sul ou norte? (LOURENÇO, 2008)

Conforme Microsoft (2008), o SQL Server impõe as seguintes restrições ao uso do tipo *geography*:

* Cada instância do tipo *geography* deve se ajustar em um único hemisfério. Sendo assim, nenhum objeto espacial maior do que um hemisfério pode ser armazenado como objeto deste tipo;
* Funções espaciais que requerem duas geometrias como parâmetros de entrada retornarão nulo caso os resultados não se ajustem em um único hemisfério. Assim ocorrerá também com a função de *STBuffer* caso seu resultado ultrapasse um único hemisfério.

Em contraste com o tipo de dados *geography,* o tipo de dados *geometry* trata os dados espaciais como se estes estivessem em um plano. Como tal, os resultados dos cálculos espaciais, tais como a distância entre pontos, são trabalhados utilizando métodos geométricos simples. Esta superfície plana é ilustrada na imagem 9 (AITCHISON, 2009).

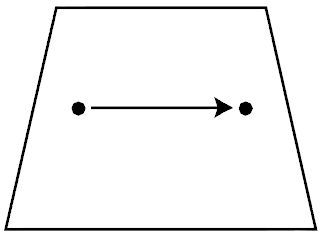


Figura 9 – Exemplo *geometry*.

Fonte: Aitchison, 2009.

Ainda, conforme Aitchison (2009), o tipo de dados *geometry* trabalha com os dados espaciais sobre uma superfície plana de duas dimensões, a posição de qualquer ponto no plano pode ser definida utilizando o armazenamento de um simples par de coordenadas cartesianas (x,y).

Dados do tipo *geometry*, quando utilizados, as coordenadas cartesianas de um ponto representam a distância do ponto de origem ao longo de um eixo definido e expressado na mesma unidade de medida das coordenadas (AITCHISON, 2009).

Uma vez que o tipo de dados *geometry* não pode considerar qualquer curvatura da Terra e não conta com a unidade de medida indicada no SRID, o fornecimento de um SRID diferente, não fará qualquer diferença no resultado final obtido (AITCHISON, 2009).

No SQL Server, o SRID corresponde a um sistema de referência espacial baseado no elipsoide específico usado para mapeamento de terra plana ou redonda. Uma coluna espacial pode conter objetos com SRIDs diferentes. No entanto, apenas instâncias espaciais com o mesmo sistema de referência podem ser usadas ao executar operações com métodos de dados espaciais (MICROSOFT, 2008).

Ambos os tipos de dados, *geometry* e *geography*, fornecem um diferente número de métodos estatísticos para a criação de objetos de dados espaciais. O método a ser utilizado dependerá da situação em que o usuário se encontra (AITCHISON, 2009).

Lourenço (2008) cita que para criar um tipo de dado espacial no SQL Server 2008, primeiramente deve-se declarar o tipo no script T-SQL, informando se é do tipo *geometry* ou *geography*. Após declarar, deve-se utilizar um método para construção de objetos. Os métodos disponíveis são:

* ***STGeomFromText*:** cria qualquer tipo de instância usando o texto de entrada;
* ***STPointFromText*:** cria uma instância *Point* usando o texto de entrada;
* ***STMPointFromText*:** cria uma instância *MultiPoint* usando o texto de entrada;
* ***STLineFromText*:** cria uma instância *LineString* usando o texto de entrada;
* ***STMLineFromText*:** cria uma instância *MultiLineString* usando o texto de entrada;
* ***STPolyFromText*:** cria uma instância *Polygon* usando o texto de entrada;
* ***STMPolyFromText*:** cria uma instância *MultiPolygon* usando o texto de entrada;
* ***STGeomCollFromText*:** cria uma instância *GeometryCollection* usando o texto de entrada.

### Operadores Espaciais

O SQL Server 2008 disponibiliza métodos definidos na OGC capazes de comparar, analisar e recuperar dados espaciais. Em função das restrições definidas pela própria ferramenta ao tipo de dados *geography*, alguns métodos estão disponíveis apenas para o tipo *geometry*. A seguir a lista dos métodos suportados e suas definições (MICROSOFT, 2008):

* ***STArea*:** retorna a área da superfície total de uma instância. Os resultados são retornados no quadrado da unidade de medida usada pelo identificador;
* ***STAsBinary*:** retorna a representação WKB do OGC de uma instância. Este método oferece suporte a instâncias espaciais maiores que um hemisfério;
* ***STAsText*:** retorna a representação WKT do OGC de uma instância. Este método oferece suporte a instâncias espaciais maiores que um hemisfério;
* ***STBoundary*:** retorna o limite de uma instância;
* ***STBuffer*:** retorna um objeto geográfico que representa a união de todos os pontos cuja distância de uma instância é menor ou igual a um valor especificado;
* ***STCentroid*:** retorna o centro geométrico de uma instância que consiste em um ou mais polígonos;
* ***STContains*:** Especifica se a instância da chamada contém espacialmente a instância passada para o método;
* ***STConvexHull*:** retorna um objeto que representa a superfície convexa de uma instância;
* ***STCrosses*:** retorna 1 se uma instância cruzar outra instância de mesmo tipo. Caso isto não ocorra, retornará 0;
* ***STDifference*:** retorna o objeto que representa os pontos de uma instância que não reside dentro de uma outra instância de mesmo tipo;
* ***STDimension*:** retorna o tipo da dimensão de uma instância;
* ***STDisjoint*:** retornará 1 se uma instância estiver espacialmente separada de outra de mesmo tipo. Caso não esteja, retornará 0;
* ***STDistance*:** retorna a distância mais curta entre um ponto em uma instância e um ponto em outra instância de mesmo tipo;
* ***STEndpoint*:** retorna o ponto de extremidade de uma instância;
* ***STEnvelope*:** retorna o retângulo delimitador mínimo alinhado pelo eixo da instância;
* ***STEquals*:** retornará 1 se uma instância representar o mesmo ponto definido como outra instância de mesmo tipo. Caso isto não ocorra, retornará 0;
* ***STExteriorRing*:** retorna o anel exterior de uma instância *geometry* que é um polígono;
* ***STGeometryN*:** retorna um elemento de uma instância especificada em uma *GeometryCollection* ou um de seus subtipos;
* ***STGeometryType*:** retorna o nome de tipo OGC representado por uma instância;
* ***STInteriorRingN*:** retorna o anel interior especificado de uma instância de *Polygongeometry*;
* ***STIntersection*:** Retorna um objeto que representa os pontos onde uma instância intercepta outra instância de mesmo tipo;
* ***STIntersects*:** retornará 1 se uma instância interceptar outra instância de mesmo tipo. Retornará 0 se isso não ocorrer;
* ***STIsClosed*:** retornará 1 se os pontos de início e término de determinada instância forem os mesmos. Retornará 1 para tipos de *geometrycollection* se cada instância contida estiver fechada. Retornará 0 se a instância não estiver fechada;
* ***STIsEmpty*:** retornará 1 se uma instância estiver vazia. Retornará 0 se uma instância não estiver vazia;
* ***STIsRing*:** retornará 1 se a instância cumprir os requisitos:
  + É uma instância *LineString*;
  + É fechada;
  + É simples.

Retornará 0 se a instância *LineString* não satisfizer os requisitos;

* ***STIsSimple*:** retorna 1 se uma instância for simples, conforme definido pelo OGC. Caso contrário, retornará 0;
* ***STIsValid*:** retornará verdadeiro se uma instância for bem formada, conforme definido pelo OGC. Caso contrário, retornará falso;
* ***STLength*:** retorna o comprimento total dos elementos de uma instância;
* ***STNumGeometries*:** retorna o número de geometrias que compõem a instância;
* ***STNumInteriorRing*:** retorna o número de anéis interiores de uma instância de *Polygongeometry*;
* ***STNumPoints*:** retorna a soma do número de pontos em cada figura de uma instância;
* ***STOverlaps*:** retornará 1 se uma instância sobrepuser outra instância de mesmo tipo*.* Se isso não ocorrer, retornará 0;
* ***STPointN*:** retorna o ponto especificado em uma instância;
* ***STPointOnSurface*:** retorna um ponto arbitrário localizado dentro de uma instância;
* ***STRelate*:** retornará 1 se uma instância tiver relação com outra instância de mesmo tipo, sendo que a relação é definida por um valor de matriz de padrão DE-9IM (matriz de 9 interseções estendida dimensionalmente). Caso contrário, retornará 0;
* ***STSrid*:** inteiro que representa o identificador de referência espacial da instância;
* ***STStartPoint*:** retorna o ponto inicial de uma instância;
* ***STSymDifference*:** retorna um objeto que representa todos os pontos que estão em uma ou mais instâncias, mas não os pontos que residem em ambas as instâncias;
* ***STTouches*:** retorna 1 se uma instância tocar espacialmente outra instância de mesmo tipo. Caso contrário, retornará 0;
* ***STUnion*:** retorna o objeto que representa a união de duas instâncias;
* ***STWithin:***retornará 1 se uma instância estiver completamente dentro de outra instância. Caso contrário, retornará 0*;*
* ***STX*:** a propriedade coordenada X de uma instância de *Point*;
* ***STY*:** a propriedade coordenada Y de uma instância de *Point*.

### Indexação Espacial

Uma indexação eficaz é a chave para que aplicações de banco de dados encontrem os resultados que o usuário deseja de forma rápida e eficiente. Considerando que os tipos de dados *geometry* e *geography* foram projetados para armazenar dados espaciais, o SQL Server 2008 também inclui um novo tipo de índice para ser usado, que se chama (talvez sem surpresa) de índice espacial (AITCHISON,2009).

Conforme Microsoft (2008), para otimizar as consultas espaciais, o SQL Server 2008 utiliza o sistema de árvore B para criar seus índices espaciais. Isso significa que os índices devem representar os dados espaciais bidimensionais na ordem linear de árvore B. Sendo assim, antes de ler dados em um índice espacial, o SQL Server 2008 implementa uma decomposição uniforme e hierárquica do espaço.

Segundo Lourenço (2008), o processo de criação de índice decompõe o espaço em uma hierarquia de grade de quatro níveis, sendo o nível um o superior. Cada nível sucessivo decompõe ainda mais o nível acima dele, de modo que cada célula do nível superior contenha uma grade completa no nível seguinte.

O SQL Server 2008, assim como outros bancos de dados espaciais, realiza a indexação definindo um sistema de grade que se estende por toda a área na qual se encontra a geometria e cada recurso a ser indexado intersecta uma ou células na grade. As células da grade são organizadas logicamente e ordenadas. A entrada do índice descreve a referência de cada recurso armazenado nas células da grade que cruza a geometria. A figura 10 ilustra uma geometria qualquer que é decomposta em todos os quatro níveis da hierarquia de uma grade 4x4 (AITCHISON, 2009).

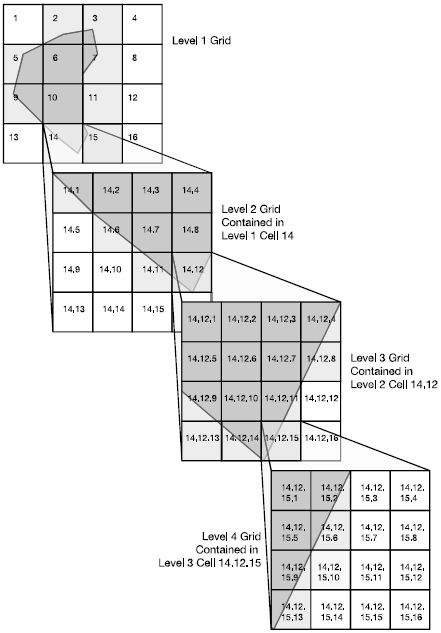


Figura 10 – Decomposição de uma geometria.

Fonte: Aitchison, 2009.

### Recurso de Carga de Dados

Os tipos de dados *geography* e *geometry* incluem métodos para importar e exportar dados nos formatos WKT (*WellKnownText*) e WKB (*WellKnownBinary*), para dados geográficos definidos pelo OGC, bem como o formato GML comumente usado, o que facilita a importação de dados geográficos de fontes que possuem suporte para estes padrões. Dados geográficos estão disponíveis em uma série de fontes governamentais e comerciais, e podem ser exportados de modo relativamente fácil de muitas aplicações SIG e sistemas GPS (*Global Position System*) existentes (MALCOLM, 2007).

Além dos formatos de carga de dados informados anteriormente, o SQL Server 2008 ainda possui uma ferramenta chamada Shape2sql que foi desenvolvida para realizar a importação de dados no formato SHP para o banco de dados. Trata-se de uma ferramenta de interface gráfica, muito simples, intuitiva e de fácil utilização por parte do usuário, mesmo os mais iniciantes conforme demonstra a figura 11 (CODEPLEX, 2010).

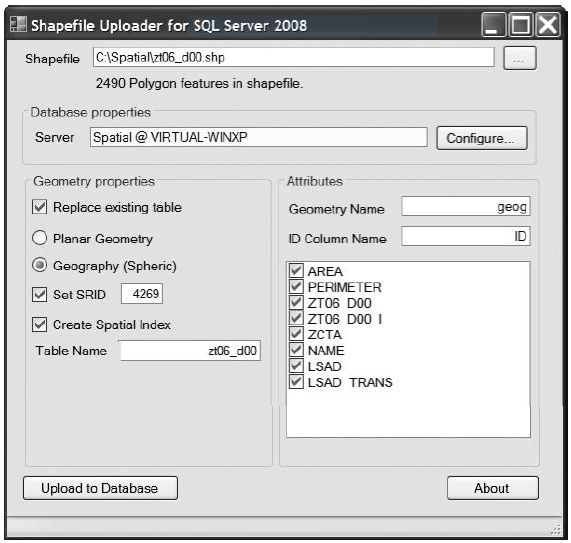


Figura 11 – Ferramenta de importação para o SQL Server 2008.

Fonte: Aitchison, 2009.

## Comparação entre os SGBDs

Após os módulos espaciais dos três SGBDs serem explicados teoricamente, um quadro comparativo entre os módulos foi desenvolvido para que as características mais significantes pudessem ser melhor observadas. O quadro 1 não distingue qual SGBD possui o melhor ou pior módulo, visto que apenas no SQL Server 2008 foram realizados experimentos práticos.



Quadro 1 – Comparação entre os módulos espaciais.

# VALIDAÇÃO DOS RECURSOS ESPACIAIS DO SQL SERVER 2008

Depois de concluir a primeira parte, a teórica, foi iniciada uma pesquisa para encontrar um estudo de caso que se encaixasse com as análises previstas para serem executadas através do SQL Server 2008.

Por se tratar de uma área muito extensa, o presente trabalho irá trabalhar na prática, com os tipos de dados espaciais do SQL Server 2008 que são *geometry* e *geography*. Mesmo com a restrição de se trabalhar apenas com o SQL Server 2008, ainda existe uma grande área de atuação e devido a isto, o foco desta pesquisa foi exclusivamente direcionado ao tipo de dados *geography*, que é o tipo de dado que leva em consideração a curvatura do Planeta Terra.

## Estudo de Caso

As regiões dos vales do Rio do Sinos e Rio Paranhana juntas somam um total de vinte municípios (quatorze no vale do Rio do Sinos e seis no vale do Rio Paranhana) e destes, conforme a figura 12, dez foram selecionados devido a proximidade de suas localizações geográficas (IBGE, 2010):

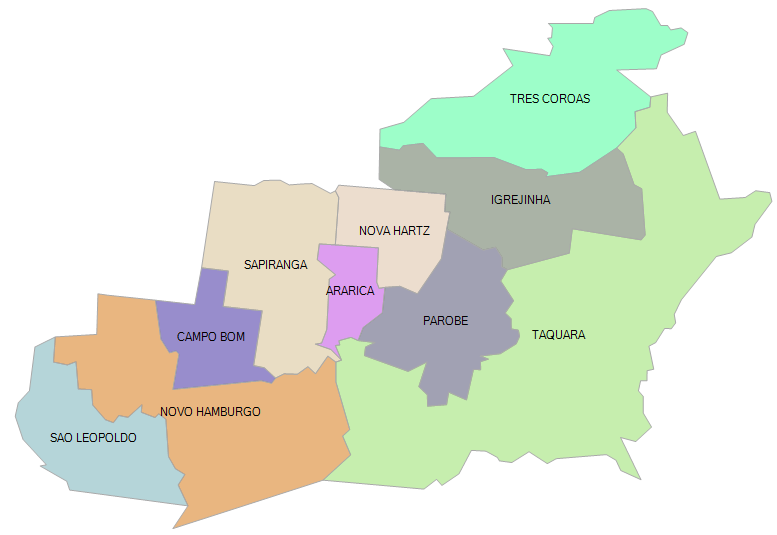


Figura 12 – Divisão atual dos municípios.

As informações municipais e os arquivos em formato SHP foram obtidos através do site do Instituto Brasileiro de Geografia e Estatística (IBGE) e são relativas ao censo do ano de 2010.

O início deste estudo de caso irá tratar sobre os municípios de Taquara e São Leopoldo. Estas duas cidades foram e ainda são muito importantes devido suas influências nos setores da indústria, agricultura, comércio, dentre outros. Juntas possuíam uma grande extensão territorial, que mais tarde originou outros vários municípios.

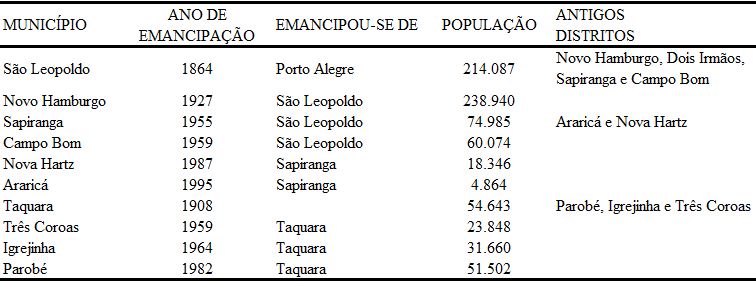
Devido a grande dificuldade de se obter dados exatos sobre as reais divisas municipais da época, para a realização deste trabalho considerou-se que a área dos municípios de Taquara em 1908 e São Leopoldo em 1864, era a soma da área de seus respectivos distritos, como ilustrado na figura 13.



Figura 13 – Antiga extensão territorial de São Leopoldo e Taquara.

A tabela 1 apresenta a relação, em ordem cronológica, dos dez municípios deste estudo de caso, contendo informações e características sobre cada um destes municípios (IBGE, 2010):

Tabela 1: Municípios utilizados no estudo de caso.



## Ferramentas Utilizadas

Para que fossem realizados testes e análises no módulo espacial do SQL Server 2008, foi utilizado um computador com processador Intel Core 2 Quad, com 8 GB (*Gigabyte*) de memória e placa de vídeo de 1 GB. Neste computador foram instaladas as seguintes ferramentas:

* **SQL Server 2008:** utilizado para criar tabelas e também permitir que consultas e funções fossem executadas;
* **Shape2Sql:** utilizado para a importação dos arquivos SHP no SGBD;
* **SqlSpatial:** utilizado para visualização gráfica dos dados através de consultas SQL e para exportar arquivos no formato SHP;
* **ArcGIS:** utilizado para a demonstração de exemplos, através de interface gráfica, na área do geoprocessamento.

## Utilização do Módulo Espacial do SQL Server 2008

Conforme detalhado anteriormente na seção 2.3.1, o SQL Server 2008 oferece suporte ao tipo de dados *geography*, que será utilizado para realizar várias operações distintas, desde a criação de tabelas, até a execução de diferentes funções específicas do módulo de operação espacial.

Todas as operações executadas no SQL Server 2008 foram realizadas através de comandos na linguagem T-SQL diretamente no banco de dados. Por meio destas consultas, foi possível a manipulação dos dados espaciais de acordo com as necessidades surgidas no decorrer da execução do trabalho em questão.

Na figura 14, é possível observar a tela inicial do SQL Server 2008, onde as consultas foram realizadas. Neste primeiro exemplo não foram utilizados dados espaciais, apenas uma consulta a uma tabela de sistema. Neste caso, existem apenas duas abas na tela de resultados.



Figura 14 – Tela inicial do SQL Server 2008.

Já na figura 15 foi realizada uma consulta em uma tabela com registros espaciais (*geography* ou *geometry*) a fim de que se pudesse verificar a diferença na tela de resultados. Pode-se perceber que quando se está manipulando dados espaciais, surge uma nova aba com a descrição “Resultados espaciais” e, além disso, pode-se adicionar as linhas latitudinais e longitudinais selecionando a opção “Mostrar linhas de grade”.

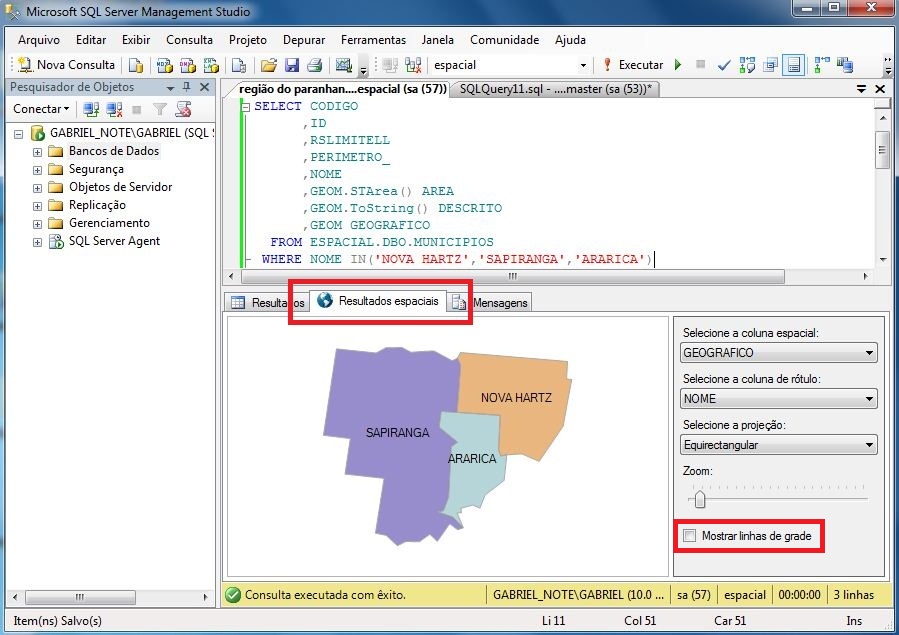


Figura 15 – Tela do SQL Server 2008 utilizando dados espaciais.

### Inserção dos Dados

Primeiramente, os arquivos em formato SHP contendo as informações de todos os municípios foram coletados diretamente do site do IBGE, onde estão disponibilizados de forma gratuita a qualquer usuário. Os arquivos neste formato armazenam informações como coordenadas latitudinais e longitudinais, código do IBGE e limites municipais.

Cada um dos municípios estudados foi inserido individualmente no SGBD SQL Server 2008, através da ferramenta Shape2Sql. A ferramenta permite configurar a conexão com o banco de dados, as colunas da tabela, SRID, índices e outras informações, como demonstra a figura 16.

A utilização do SRID de código 4326 se dá pelo fato deste ser o código padrão utilizado pelo SQL Server 2008 e é mapeado pelo sistema de referência espacial WGS84 (*World Geodetic System*) que, segundo Silva (2010), utiliza como referência o Elipsoide Internacional de 1980 (GRS-80). (MICROSOFT, 2008).

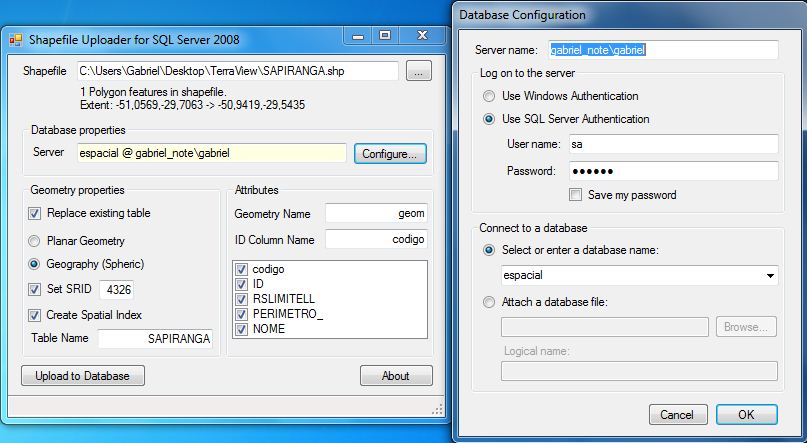


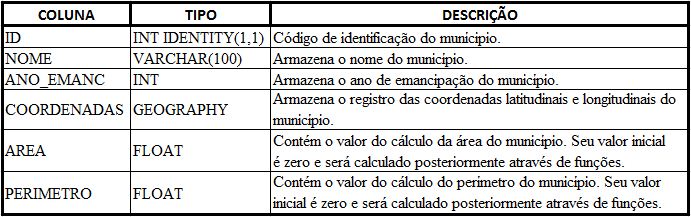
Figura 16 – Inserção dos dados no SGBD.

Os arquivos SHP foram inseridos em tabelas provisórias criadas automaticamente pela ferramenta Shape2Sql. Os nomes destas tabelas foram definidos conforme o preenchimento do campo *Table Name* (nome da tabela), e já possuem algumas informações em seu interior. As informações das colunas em que se encontram o nome e as coordenadas geográficas foram mantidas para que mais tarde fossem transferidas para uma nova tabela criada especificamente para armazenar cada um dos municípios. Demais tabelas também foram criadas para auxiliar na manipulação dos dados.

Com os arquivos já inseridos no SGBD e, por enquanto, sem nenhuma alteração, foi possível a visualização dos municípios, exibidos na figura 12, dispostos conforme estão nos dias de hoje.

### Criação e Manipulação das Tabelas

Para a realização desta próxima etapa, uma nova tabela, de nome MUNICIPIOS, foi criada a fim de armazenar todos os dez municípios individualmente. Para esta tabela foram transferidas as informações de coordenadas e nomes dos munícipios que se encontravam nas tabelas provisórias e adicionadas novas colunas, ficando a tabela MUNICIPIOS conforme o quadro 2.



Quadro 2 – Descrição da tabela MUNICIPIOS.

Os dados armazenados em colunas espaciais, quando utilizados em instruções T-SQL comuns, possuem seus resultados exibidos de forma codificada (sinalizado em azul na figura 17). Devido a isto, a função *ToString()* foi utilizada, para que estes dados codificados pudessem ser exibidos de forma legível ao usuário (sinalizado em vermelho na figura 17) e este poder utilizá-los para a formação dos novos pontos geográficos.

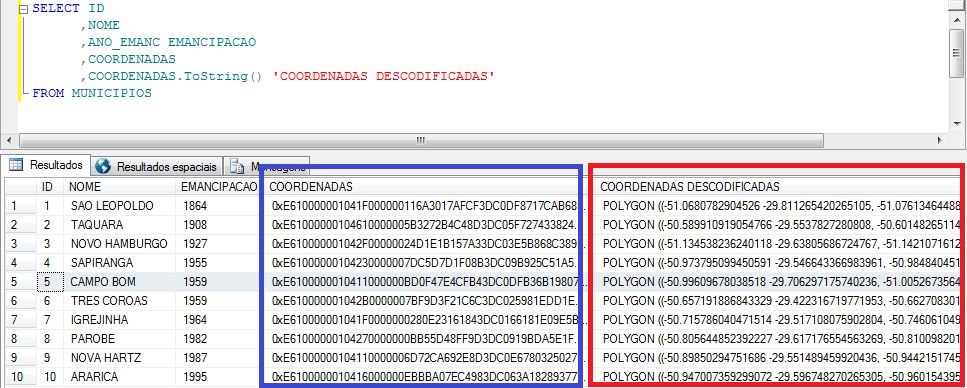


Figura 17 – Representação de dados codificados e não codificados.

Outras duas funções que também podem ser utilizadas para a codificação e decodificação de coordenadas são *STAsText()*, que transforma os dados binários em dados legíveis iguais ao da função *ToString()* e *STAsBinary()*, que faz o processo inverso, transformando os dados legíveis em binários.

Para que as coordenadas dos municípios existentes nas tabelas provisórias fossem copiadas e inseridas na tabela MUNICIPIOS, duas formas diferentes foram utilizadas. A primeira foi através da criação de uma variável que armazena a coordenada (codificada) e depois a insere na tabela MUNICIPIOS. A segunda foi através da utilização da função *STPolyFromText()*, onde foi possível a visualização das coordenadas decodificadas que foram copiadas e então inseridas através de uma instrução *INSERT* como demonstra a figura 18.

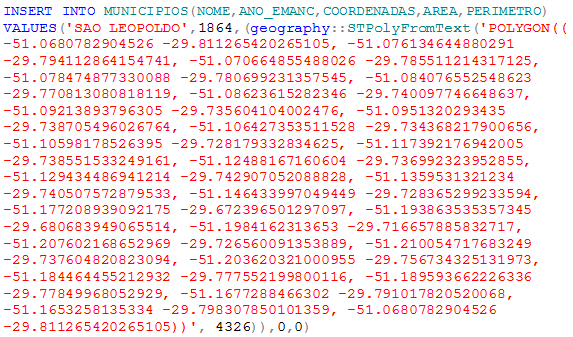
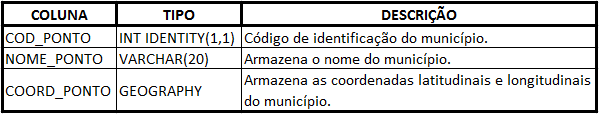


Figura 18 – Inserção de coordenadas decodificadas.

O próximo passo foi o de criar a tabela MUNICIPIOS\_ANTIGOS, mostrada no quadro 3, para armazenar os municípios de São Leopoldo e Taquara conforme os anos de 1864 e 1908 respectivamente. Através desta nova tabela serão demonstrados dois possíveis métodos de formação e inserção dos dois municípios antigos.

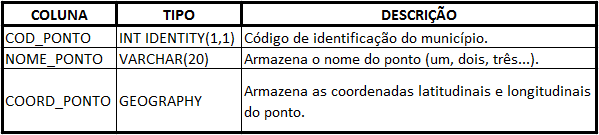


Quadro 3 – Descrição da tabela MUNICIPIOS\_ANTIGOS.

Com a tabela MUNICIPIOS\_ANTIGOS criada, o próximo passo foi separar as coordenadas do município de Taquara e seus três distritos, que foram inseridas como polígono, e adicioná-las novamente em outra tabela provisória, só que como ponto.

Com a visualização do município de Taquara e seus três distritos em formato de pontos, tornou-se mais fácil a identificação de todos os pontos da divisa territorial entre os municípios do vale do Paranhana. Então, todos os pontos pertencentes as divisas internas (que fazem divisa com Taquara e entre os outros municípios do vale do Rio Paranhana) foram retirados, deixando somente os pontos externos, formando assim o município de Taquara no ano de 1908.

Para que fosse possível realizar o levantamento de quais pontos seriam necessários, foram criadas quatro (município de taquara e seus respectivos distritos) tabelas de nome VERIFICA\_TAMANHO\_MUNICIPIO\_<NOME>, onde <NOME> é alterado conforme o nome do município em questão, e nestas foram inseridos cada um dos municípios do vale do Paranhana em formato de pontos, conforme demonstra o quadro 4.



Quadro 4 – Descrição da tabela VERIFICA\_TAMANHO\_MUNICIPIO\_<NOME>.

A operação de inserção dos pontos nas quatro tabelas VERIFICA\_TAMANHO\_MUNICIPIO\_<NOME> foi realizada com a utilização da função, que permite a inserção de pontos, *STPointFromText()* (figura 19).



Figura 19 – Inserção de pontos para a criação do antigo município de Taquara.

Estando as quatro tabelas VERIFICA\_TAMANHO\_MUNICIPIO\_<NOME> com os municípios inseridos em formato de pontos, a próxima etapa foi a de analisar cada um dos pontos e verificar os que deveriam ou não fazer parte da inserção de dados na tabela MUNICIPIOS\_ANTIGOS, que armazenará as extensões territoriais de São Leopoldo e Taquara nos anos de 1864 e 1908. Abaixo, na figura 20, é possível observar um exemplo do município de Três Coroas em formato de pontos na tabela VERIFICA\_TAMANHO\_MUNICIPIO\_TRES\_COROAS.



Figura 20 – Município de Três Coroas em formato de pontos.

### Funções Espaciais e Análise dos Dados

Depois de analisados, os pontos (coordenadas) selecionados foram adicionados em um comando de instrução *INSERT* a fim de formar um único polígono representando o município de Taquara. Neste processo, a função *STPolyFromText()* foi utilizada novamente para que o município em questão fosse inserido na tabela MUNICIPIOS\_ANTIGOS.

Com a tabela MUNICIPIOS\_ANTIGOS já contendo o antigo município de Taquara, é chegada a vez de o município de São Leopoldo também ser inserido na mesma. Para isto, a função *STUnion()* será utilizada, mostrando assim, mais de uma forma de se realizar mesmas operações.

Na figura 21 é demonstrada a consulta que foi utilizada para a criação do município de São Leopoldo no ano de 1864. Nesta consulta foram declaradas (*declare*) variáveis e, posteriormente, estas foram preenchidas (*set*) com as coordenadas de seu respectivo município. Depois disto, o primeiro município é unido (*STUnion()*) com o segundo, que por sua vez, é unido com o terceiro e assim sucessivamente.

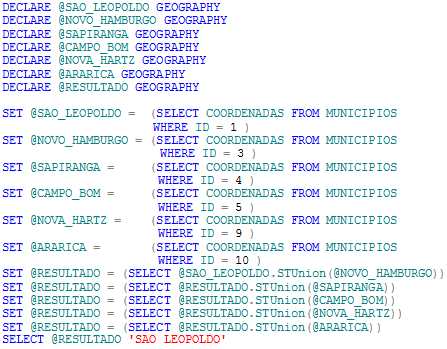


Figura 21 – Inserção com a utilização da função *STUnion()*.

Após o município de São Leopoldo e seus cinco distritos estarem unidos pela função *STUnion()*, foi realizada a inserção desta união na tabela MUNICIPIOS\_ANTIGOS, onde já se encontrava o município de Taquara. Para que esta inserção fosse possível, utilizou-se a variável @resultado (consulta anterior) em um novo comando *INSERT*, demonstrado na figura 22.



Figura 22 – Inserção através de variável com *STUnion()*.

Com a utilização da função *STUnion()*, a integração entre os municípios e seus distritos se tornou muito mais rápida e prática. Abaixo, na figura 23, é mostrada a tabela MUNICIPIOS\_ANTIGOS e o resultado da união dos dois municípios.



Figura 23 – Municípios de São Leopoldo e Taquara nos anos de 1864 e 1908, respectivamente.

Caso o usuário receba um arquivo SHP criado em outro local e com um SRID diferente, possivelmente será gerado um erro quando este for manipulado em operações com os arquivos já existentes. Para evitar este problema, existe a função *STSrid()*, demonstrada na figura 24, que ao concatená-la com a coluna do tipo *geography*, exibe o SRID dos dados (por exemplo, 4326).

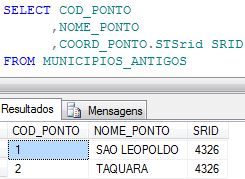


Figura 24 – Exibição do SRID.

Com a utilização da função *STArea()* é possível obter o tamanho total de uma determinada área (polígono). Esta função disponibilizará o retorno em metros quadrados. Na consulta, exibida na figura 25, foram calculadas as áreas dos municípios de São Leopoldo e Taquara.

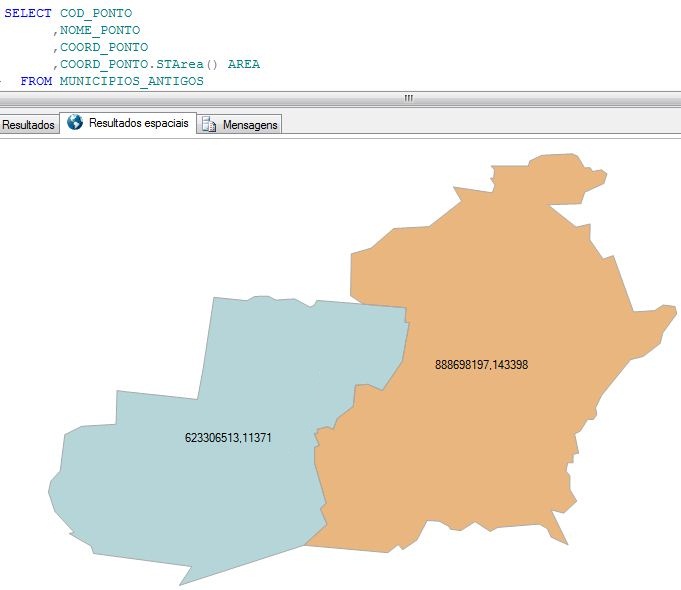
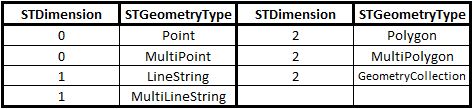


Figura 25 – Cálculo de área através da função *STArea()*.

Há também a função *STDimension()* que mostra o tipo de dimensão que está sendo utilizado por cada um dos registros da tabela. O retorno desta função é o código, que vai de zero à dois, definindo as dimensões, conforme o quadro 5.



Quadro 5 – Legendas das funções *STDimension()* e *STGeometryType()*.

A função *STGeometryType()* possui muitas semelhanças com a *STDimension()*, pois também retorna a dimensão utilizada, porém, não em forma de código e sim em forma de texto, conforme demonstrado no quadro 5.

Na figura 26, as coordenadas das atuais prefeituras de São Leopoldo e Taquara foram inseridas como ponto para que pudesse ser vista a utilização e difenças das duas funções mencionadas no quadro 2.

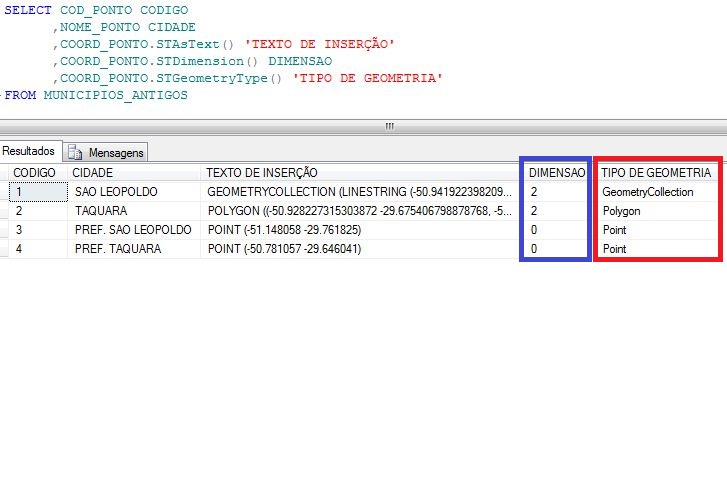


Figura 26 – Utilização das funções *STDimension()* e *STGeometryType().*

Conforme demonstrado na figura 26, ambos os municípios (São Leopoldo e Taquara) possuem dimensão dois, porém São Lopoldo é de uma geometria *GeometryCollection* e Taquara *Polygon*. Isto ocorre devido ao fato de ambos os municípios terem sido criados de formas diferentes.

O município de Taquara foi criado através do sistema de transformação de polígonos em pontos e então foram selecionados os pontos que deveriam ou não ficar. Após isto, os pontos que restaram foram inseridos em forma de polígono na tabela MUNICIPIOS\_ANTIGOS.

Já o município de São Leopoldo foi criado através da união (*STUnion()*) de vários outros polígonos, que por sua vez, após estarem devidamente unidos, foram inseridos, todos juntos, na tabela MUNICIPIOS\_ANTIGOS, formando assim uma coleção de polígonos.

Com a tabela MUNICIPIOS\_ANTIGOS já contendo os municípios de Taquara e São Leopoldo unidos a seus distritos, é chegada a hora de se voltar aos dias atuais utilizando a tabela MUNICIPIOS que armazena em seus registros informação de todos os municípios em questão neste trabalho.

Na tabela MUNICIPIOS, as colunas de nomes AREA e PERIMETRO ainda não tiveram seus registros inseridos. Estes registros serão inseridos em ordem de emancipação dos municípios e a primeira coluna a ser preenchida será a da AREA. Com a inserção de todos os registros na tabela, comparações e análises poderão ser realizadas e assim, diferenças entre passado e presente poderão ser melhores visualizadas.

Conforme a figura 27, a inserção dos dados na coluna AREA foi realizada através de um *cursor* e a função *STArea()* foi novamente utilizada.

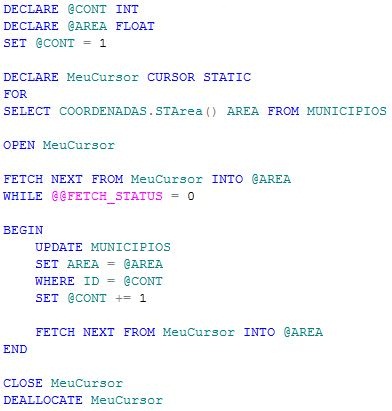


Figura 27 – Utilização de *cursor* para inserir dados na coluna AREA.

Com a coluna AREA preenchida, para que a tabela municípios ficasse completa, faltava apenas inserir os dados na coluna PERIMETRO. Para realizar esta operação, a função *STLength()* foi utilizada.

A função *STLength()*, quando aplicada em uma instância qualquer, retorna o seu comprimento em metros. Se for utilizada em uma *Linestring* (linha), retornará o seu comprimento, em um *Point* (ponto), retornará o valor de zero e em um *Polygon* (polígono), irá retornar o valor de seu perímetro (soma de todos os lados).

O preenchimento da coluna PERIMETRO, assim como o da AREA, se deu através da utilização de um *cursor*, muito semelhante ao exibido na consulta anterior, porém com diferenças nos nomes das colunas e variáveis. Com a conclusão do preenchimento dos dados que faltava, a tabela MUNICIPIOS, conforme a figura 28 está completa.

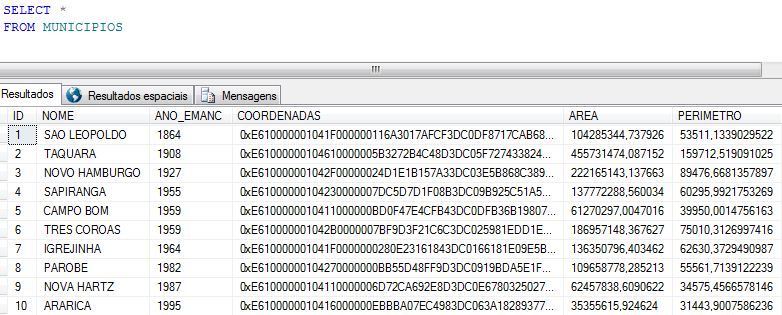
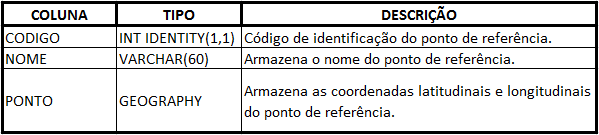


Figura 28 – Exibição completa da tabela MUNICIPIOS.

Estando a tabela de MUNICIPIOS totalmente preenchida, foi possível visualizar, no pior caso, a grande perda de extensão territorial por parte do município de São Leopoldo que de 623,306 Km² em 1864, demonstrado na figura 25, restaram apenas 102,739 Km² em 2010.

Para a execução de mais demonstrações e análises, foi criada a tabela PONTOS, exibida no quadro 6, e algumas coordenadas nela foram inseridas, a começar com a posição atual de todas as dez prefeituras e alguns pontos centrais em distritos ainda existentes.



Quadro 6 – Descrição da tabela PONTOS.

A figura 29 apresenta os municípios com os pontos de localização de suas prefeituras. Em cada um dos distritos dos municípios de Parobé e Novo Hamburgo, também existe um ponto central indicando o centro deste distrito, exceto no distritos de Taquara, aonde existem dois pontos centrais devido a sua extensão territorial. Para facilitar a visualização, nos pontos referentes a localização das prefeituras foram adicionadas as siglas PFT e nos pontos referentes aos centros dos distritos foram inseridas as siglas DTS.

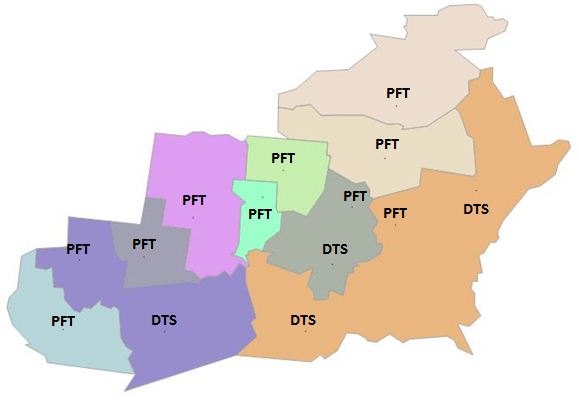


Figura 29 – Posição das prefeituras e alguns de seus distritos.

Os distritos inseridos são Rio da Ilha e Fazenda Fialho no município de Taquara, Lomba Grande no município de Novo Hamburgo e Santa Cristina do Pinhal no município de Parobé.

A próxima função utilizada foi a função *STBuffer()*. Nesta função é declarada uma determinada instância (ponto, linha ou polígono) que será manipulada e então é passado como parâmetro a área (em metros) de abrangência que esta instância irá atingir. A figura 30 demonstra as prefeituras e os distritos dos municípios de Taquara, Novo Hamburgo e Parobé recebendo um *Buffer* de cinco mil metros de abrangência através da união (*union all)* das tabelas MUNICIPIOS e PONTOS.

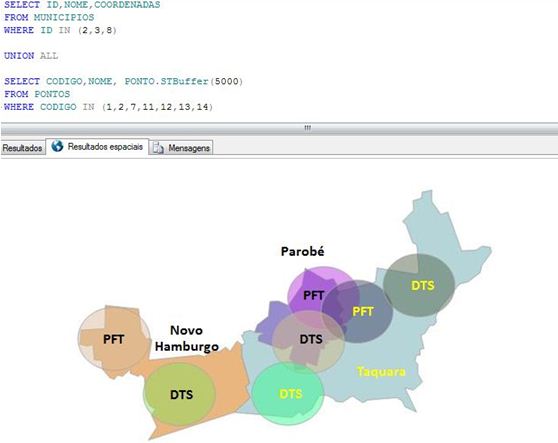


Figura 30 – Prefeituras e distritos com um *Buffer* de cinco mil metros de abrangência.

Conforme a figura 30, pode-se identificar que ao ser utilizado um *Buffer* de cinco mil metros a área urbana das cidades selecionadas ficaram quase que 100% cobertas. Com isso, é visível que os municípios de Taquara e Novo Hamburgo ainda possuem uma grande extensão territorial hoje caracterizada como “zona rural”.

Para que estas diferenças e distâncias dos municípios de Taquara e Novo Hamburgo fiquem mais claras, utilizou-se a função *STDistance()* que realiza o cálculo da distância entre duas instâncias através de uma linha reta e disponibiliza seu resultado em metros.

Na figura 31 é demonstrada a distância entre as prefeituras dos municípios de Taquara e Novo Hamburgo e o centro de seus respectivos distritos.

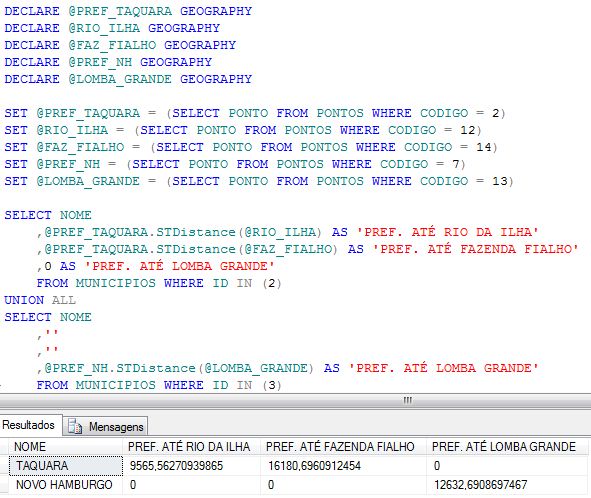


Figura 31 – Aplicação da função *STDistance().*

Através da figura 31 foi identificado (nos dois piores casos) que a prefeitura de Taquara está, em sua maior distância, a aproximadamente dezesseis quilômetros de distância do centro do distrito de Fazenda Fialho e a prefeitura de Novo Hamburgo a aproximadamente doze quilômetros de distância de seu distrito Lomba Grande.

Com a utilização das figuras 30 e 31, é possível pensar em questões do tipo:

* Por que os distritos de Fazenda Fialho e Lomba Grande não se emancipam?
* Será que os distritos de Fazenda Fialho e Lomba Grande possuem menor infra estrutura do que , por exemplo, o município de Araricá?

Porém, para um município se emancipar não conta somente a sua extensão territorial, pois se fosse assim, municípios como Bom Jesus, Caxias do Sul ou São Francisco de Paula já deveriam ter suas extensões territoriais reduzidas. Nestes casos, todo um estudo de viabilidade deve ser realizado antes de se tomar alguma atitude.

### Outras Funções Espaciais

Como o SQL Server 2008 possui ainda várias outras funções, as funções apresentadas a seguir serão apenas de caráter informativo, não sendo realizado nenhum tipo de análise específica sobre as mesmas.

Para exemplificar a função *STDifference()*, o município de São Leopoldo (antigo), que foi unido com seus distritos através da função *STUnion()*, foi comparado com o município de Novo Hamburgo. Ao ser aplicada, na figura 32, a função *STDifference()* retornou o que não havia de comum entre os dois municípios.

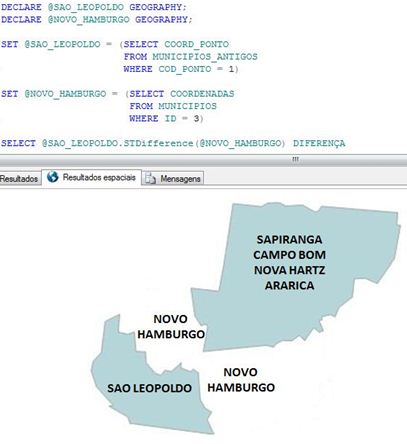


Figura 32 – Exemplo de utilização da função *STDifference()*.

A execução do processo inverso ao da função *STDifference()* é realizado com a utilização da função *STIntersection()*, exibida na figura 33, que retorna apenas o que existe de comum entre os dois municípios, que neste exemplo é o município de Novo Hamburgo.

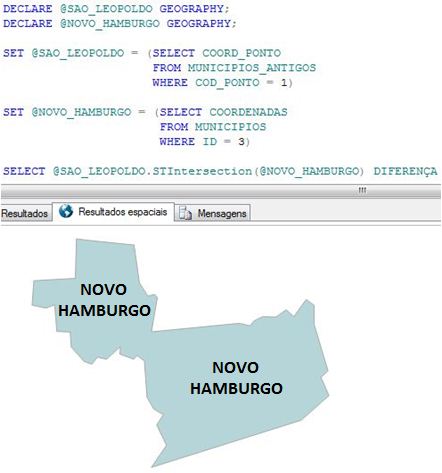


Figura 33 – Demonstração de uso da função *STIntersection()*.

Se por acaso, em alguma situação qualquer, o usuário utilizar dados espaciais que não são de seu total conhecimento, várias situações diferentes podem vir a surgir, como por exemplo:

* Saber se uma determina instância cruza ou não sobre a outra;
* Saber qual é a coordenada de início ou fim de um ponto linha ou polígono;
* Saber qual é número de pontos existentes em uma determinada instância.

Para que algumas destas situações sejam resolvidas sem mais complicações, o SQL Server 2008 possui ainda algumas outras funções que podem vir a ser muito úteis em casos como os descritos acima.

*STEquals()* é uma função tanto interesante quanto perigosa de ser utilizada. Ela é interessante pelo o que realiza e perigosa pelo o que pode esconder. Se um polígono, uma linha e um ponto são desenhados através da função *STGeomCollFromText()*, demonstrados na figura 34, e um outro polígono, de igual tamanho, porém sem a linha e o ponto, é desenhado pela função *STPolyFromText()*, conforme a figura 35, quando chamada, como demonstra a figura 36, a função *STEquals()* sinalizará que as duas instâncias são iguais, o que de fato não é verdadeiro.

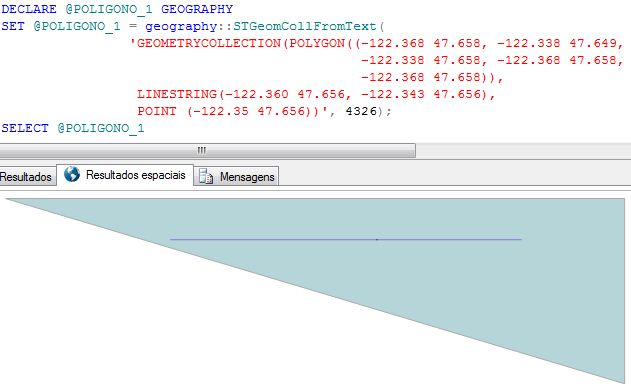


Figura 34 – *Geometry Collection* com ponto, linha e polígono.

Fonte: Microsoft, 2008.



Figura 35 – Polígono.

Fonte: Microsoft, 2008.

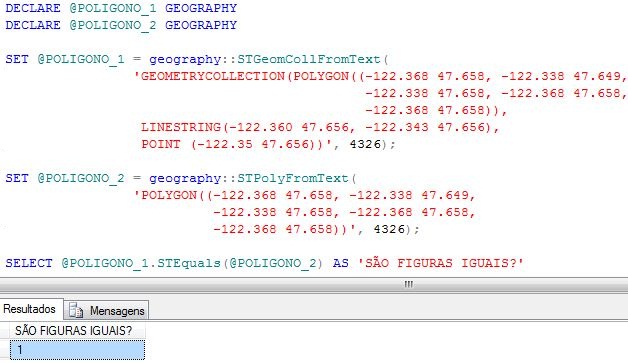


Figura 36 – Demonstração de uso da função *STEquals()*.

Fonte: Microsoft, 2008.

O que faz a função *STEquals()* retornar que as duas instâncias são iguais é o fato de que mesmo a primeira, exibida na figura 34, contendo um ponto e uma linha a mais que a segunda, exibida na figura 35, este ponto e esta linha não ultrapassam os limites exteriores do polígono e com isso ambas as instâncias são consideradas iguais.

Caso o usuário deseje saber se uma determinada instância está completamente fechada, utiliza-se a função *STIsClosed()*. Esta função ira retornar um se a instância estiver fechada ou, caso contrário, irá retornar zero. Porém, *STIsClosed()* não reconhece o fechamento correto de uma instância que foi criada através de uma *Geometry Collection*, como demonstrado na figura 34. Sendo assim, mesmo os limites exteriores das instâncias, nas figuras 34 e 35 sendo iguais, o ponto e a linha em seu interior faz com que a função não reconheça seu correto fechamento, como exibido na figura 37.

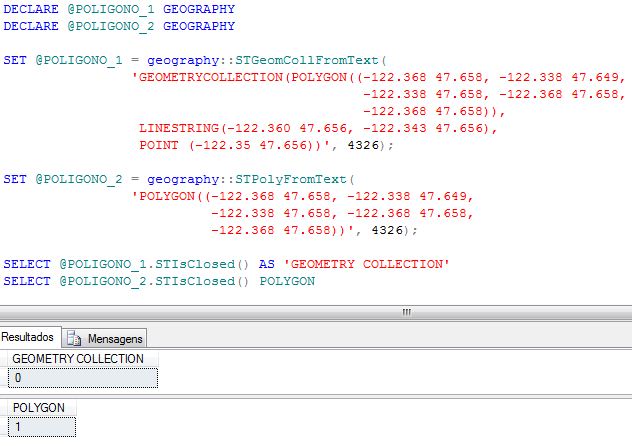


Figura 37 – Utilização da função *STIsClosed()*.

Fonte: Microsoft, 2008.

Quando se existe a necessidade em saber se uma determinada instância cruza ou faz divisa com outra, as funções *STDisjoint()* e *STIntersects()*, exibidas na figura 38, podem ser utilizadas a fim de facilitar esta identificação.

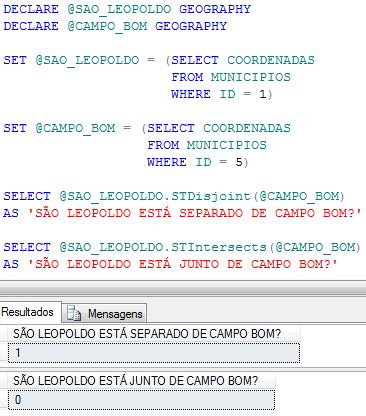


Figura 38 – Demonstração das funções *STDisjoint()* e *STIntersects()*.

A função *STDisjoint()* indica se duas ou mais instâncias estão separadas e a função *STIntersects()* indica se duas ou mais instâncias estão juntas. No exemplo da figura 38, é indicado que o município de São Leopoldo está separado (*STDisjoint()*) ou não está junto (*STIntersects()*) ao município de Campo Bom.

Em uma instância qualquer, para que seja verificado quais são os pontos iniciais ou finais da mesma, pode-se utilizar as funções *STEndPoint()* (último ponto) e *STStartPoint()* (primeiro ponto).

Na figura 39, uma instância, em forma de linha, é formada através dos conjuntos de pontos de coordenadas (0 0), (2 2), (3 1). Ao ser utilizada, a função *STEndPoint()* irá retornar o ponto formado pelas coordenadas (3 1) e ao utilizar a função *STStartPoint()*, a mesma irá retornar o ponto formado pelas coordenadas (0 0). As linhas de grade e a função *STBuffer()* foram utilizadas para que houvesse uma maior facilidade na identificação dos dois pontos.

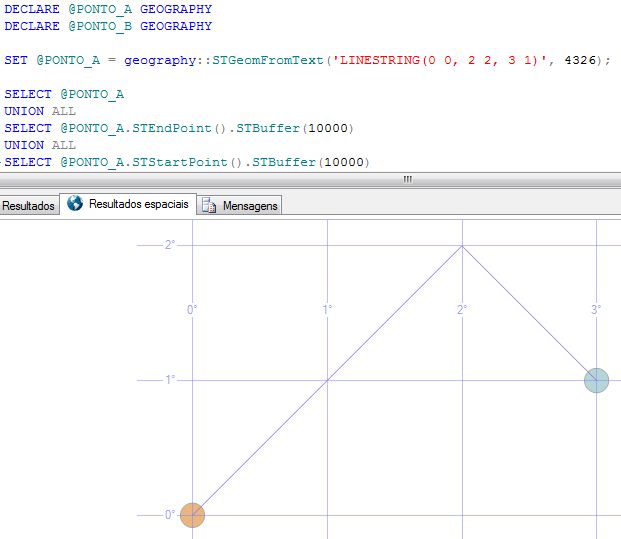


Figura 39 – Identificação dos pontos inicial e final de uma instância.

Para finalizar os experimentos sobre o tema da evolução dos municípios através de funcionalidades do módulo espacial da ferramenta SQL Server 2008, algumas funções foram reunidas para serem exemplificadas.

A figura 40, reúne cinco funções espaciais suportadas pelo tipo de dados *geography*. Cada uma delas foi tratada separadamente e todas estão relacionadas com a *Geometry Collection* (ponto, linha e polígono) ilustrada ao lado dos resultados.

A primeira função é a *STGeometryN()* que mostra as coordenadas utilizadas na instâcia especificada, neste caso, um ponto (*point*). Na sequência, a função *STPointN()* exibe a coordenada solicitada, na ordem em que as mesmas foram inseridas. No exemplo a coordenada de número seis é a primeira coordenada da *linestring*. *STNumGeometries()* é a próxima função e ela retorna o número total de instâncias existentes na *Geometry Collection*, que é três. A quarta função utilizada é a função *STNumPoints()* que demonstra que existem oito pontos no total da *Geometry Collection*. Por fim, a função *STIsEmpty()*, que retorna zero, identifica que a instância não se encontra vazia.

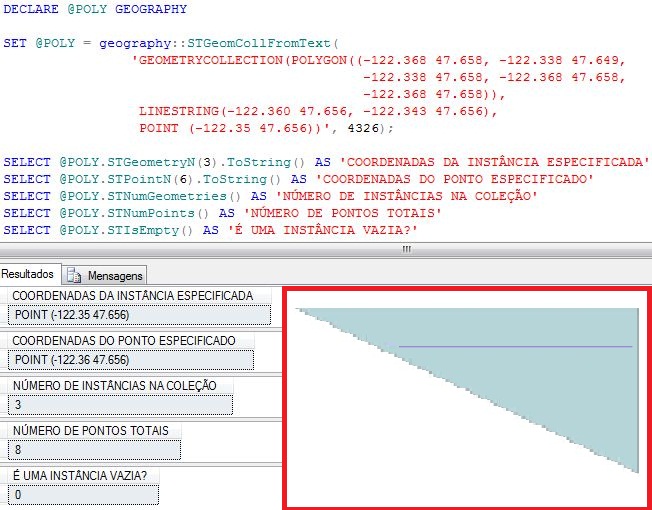


Figura 40 – Demais funções do tipo de dados *geography*.

Outra maneira de se utilizar os dados em formato SHP, é através da utilização da ferramenta SqlSpatial. Esta ferramenta possibilita que os dados contidos em arquivos do tipo *geography* ou *geometry* sejam exportados do SQL Server e utilizados em qualquer ferramenta SIG. Na figura 41, foi exibido um exemplo de consulta utilizando a função *STBuffer()* e demarcado na cor vermelha está o menu *Tools* (ferramentas) com a opção “*Export result to shapefile”* (Exportar resultado para *shapefile*).

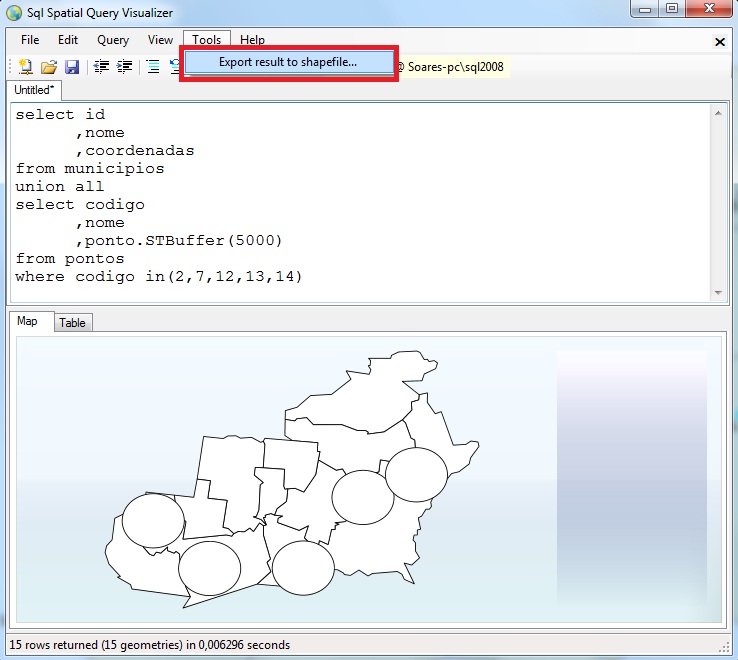


Figura 41 – Exportar dados para SHP.

## Exemplos em Ferramenta de Geoprocessamento

Após os dados terem sido manipulados no módulo espacial do SQL Server 2008, alguns exemplos de manipulação serão criados na ferramenta SIG de nome ArcGIS 10.1 for Desktop que foi desenvolvida pela empresa Esri.

O ArcGIS, segundo Esri (2012), é uma plataforma para criação, edição e análise geográfica. Possui a finalidade de examinar e testar dados, possibilitando com isto que o usuário consiga tomar a decisão correta em diferentes situações. É formado por um conjunto de ferramentas, dentre as quais, as mais utilizadas são:

* **ArcMap:** Utilizado para criação, visualização e edição de arquivos ou mapas espaciais;
* **ArcCatalog:** Utilizado para o armazenamento da coleção de arquivos ou mapas, estando estes em pastas ou através de uma conexão com um SGBD, para que fiquem disponíveis ao ArcMap;
* **ArcSDE:** Realiza a conexão com o SGBD para que os dados sejam disponibilizados no ArcCatalog.

Ainda, conforme Esri (2012), além de se conectar com o SQL Server 2008, o ArcGIS também permite que a conexão com outros SGBDs seja estabelecida. A figura 42 ilustra todos os módulos espaciais dos SGBDs que podem ser conectados ao ArcGIS.

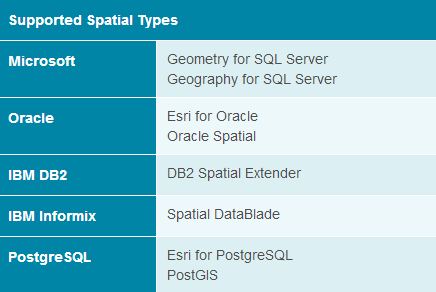


Figura 42 – Tipos de SGBDs suportados pelo ArcGIS.

Fonte: Esri, 2012.

Depois de o ArcGIS estar instalado, o ArcSDE, que é um sistema de gerenciamento de banco de dados relacional, foi a primeira ferramenta a ser utilizada. Conforme demonstra a figura 43, primeiramente foi necessária a configuração do ArcSDE para que este pudesse realizar a função de tornar o banco de dados visível ao ArcCatalog.

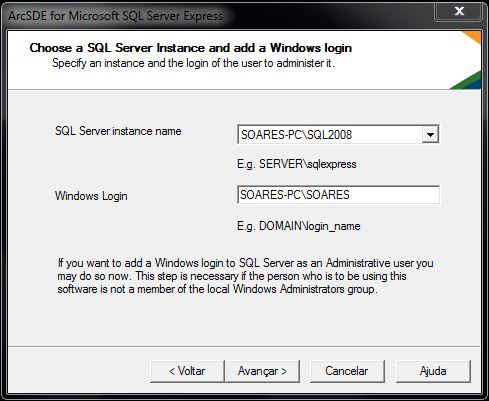


Figura 43 – Configuração de conexão com o SGBD.

Com o ArcSDE configurado, a próxima ferramenta utilizada foi o ArcCatalog, exibido na figura 44. Neste, além da conexão com o SGBD SQL Server 2008 ser estabelecida (sinalizado em vermelho na figura 44), também foi criada uma pasta para o armazenamento dos dados manipulados localmente (sinalizado em azul na figura 44), que serão exportados do SQL Server 2008 pela ferramenta SqlSpatial.

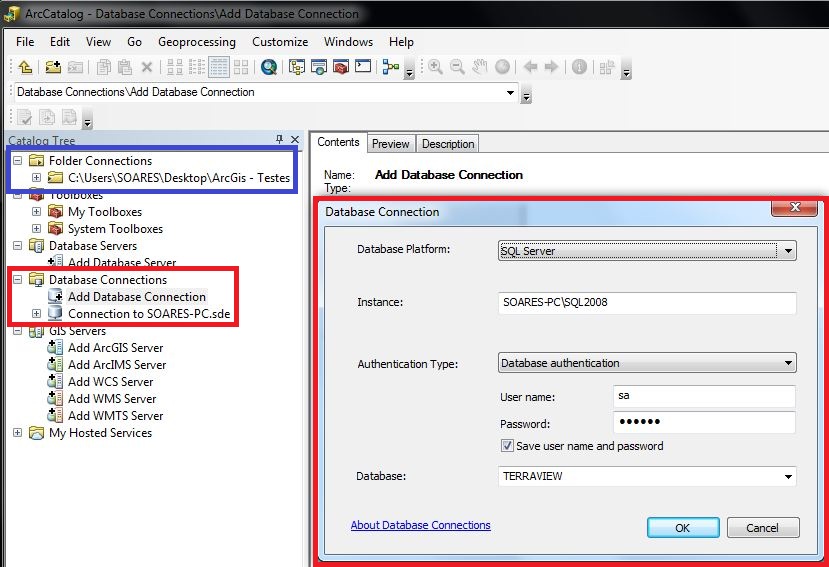


Figura 44 – Conexão com o SGBD e armazenamento de arquivos.

Conforme a figura 45, depois do ArcSDE e o ArcCatalog estarem configurados, as tabelas utilizadas anteriormente (MUNICIPIOS, PONTOS, etc.) ficaram disponíveis para serem utilizadas no ArcMap.

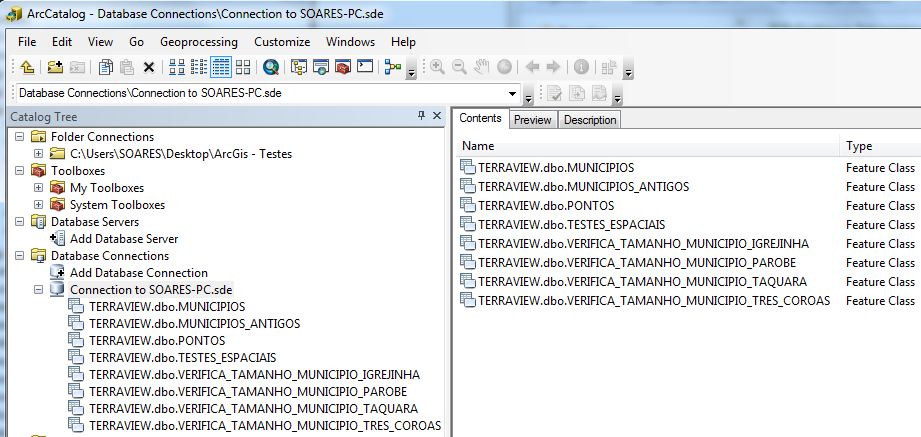


Figura 45 – Visualização das tabelas da conexão com o SGBD.

Estando as tabelas disponíveis, a ferramenta ArcMap foi utilizada para que fossem iniciadas as manipulações e análises sobre os dados. Na figura 46, é demonstrada a tela inicial do ArcMap.

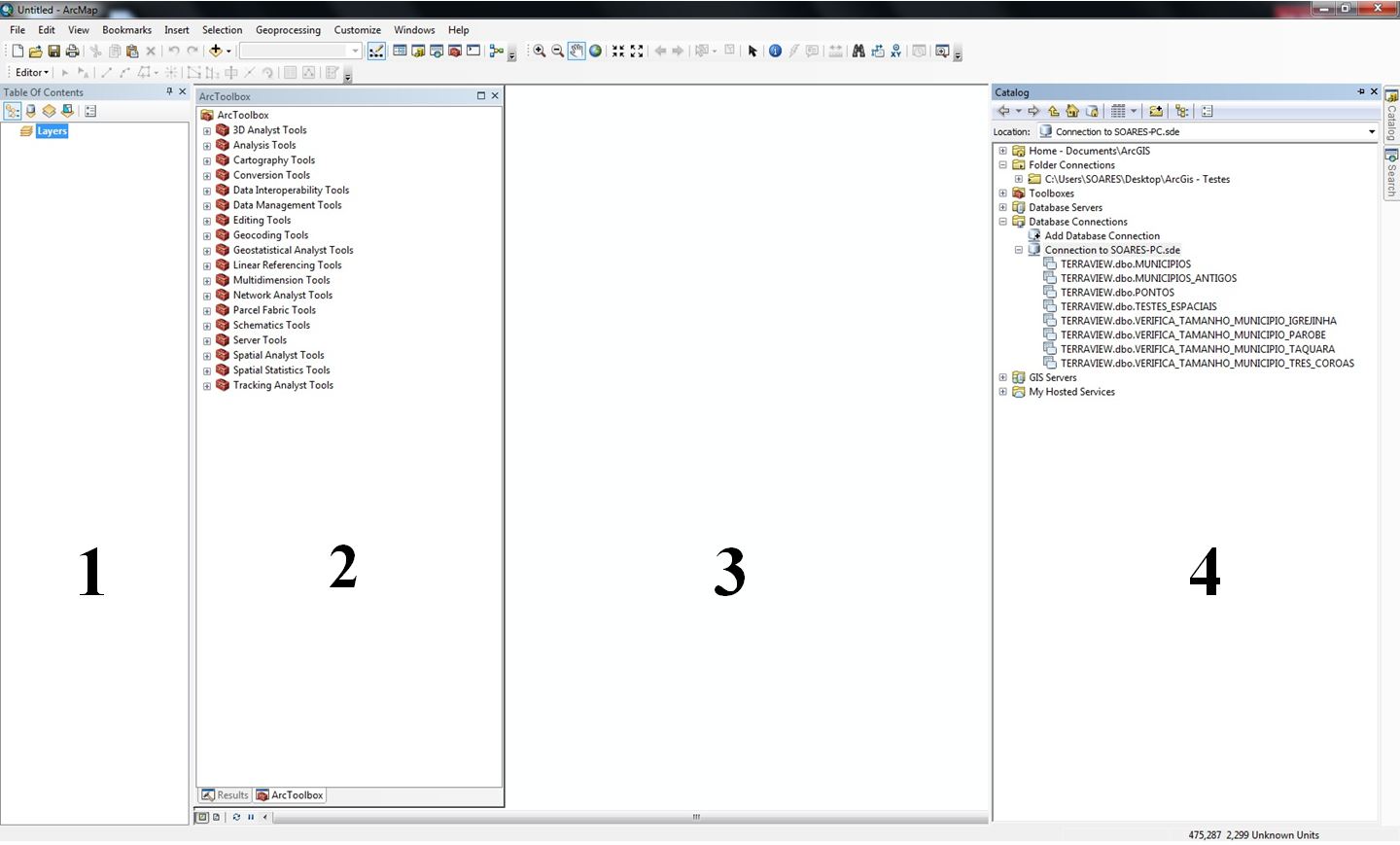


Figura 46 – Tela inicial do ArcMap.

A figura 46 foi dividida em 4 setores para que a tela inicial do ArcMap fosse melhor explicada e sua divisão segue conforme a descrição:

1. Local onde são exibidas todas as *Layers* (camadas) do projeto;
2. Local onde são exibidas todas as funções ou ferramentas possíveis para serem utilizadas;
3. Local onde os resultados de manipulações e análises são exibidos;
4. Local onde estão as tabelas do SGBD e as pastas locais para o armazenamento dos arquivos.

Nos exemplos que seguem, foram utilizadas as tabelas MUNICIPIOS e PONTOS. O primeiro passo, conforme a figura 47, foi selecionar as duas tabelas na aba do ArcCatalog e adicioná-las na camada de *Layers* para que a manipulação dos dados fosse iniciada.

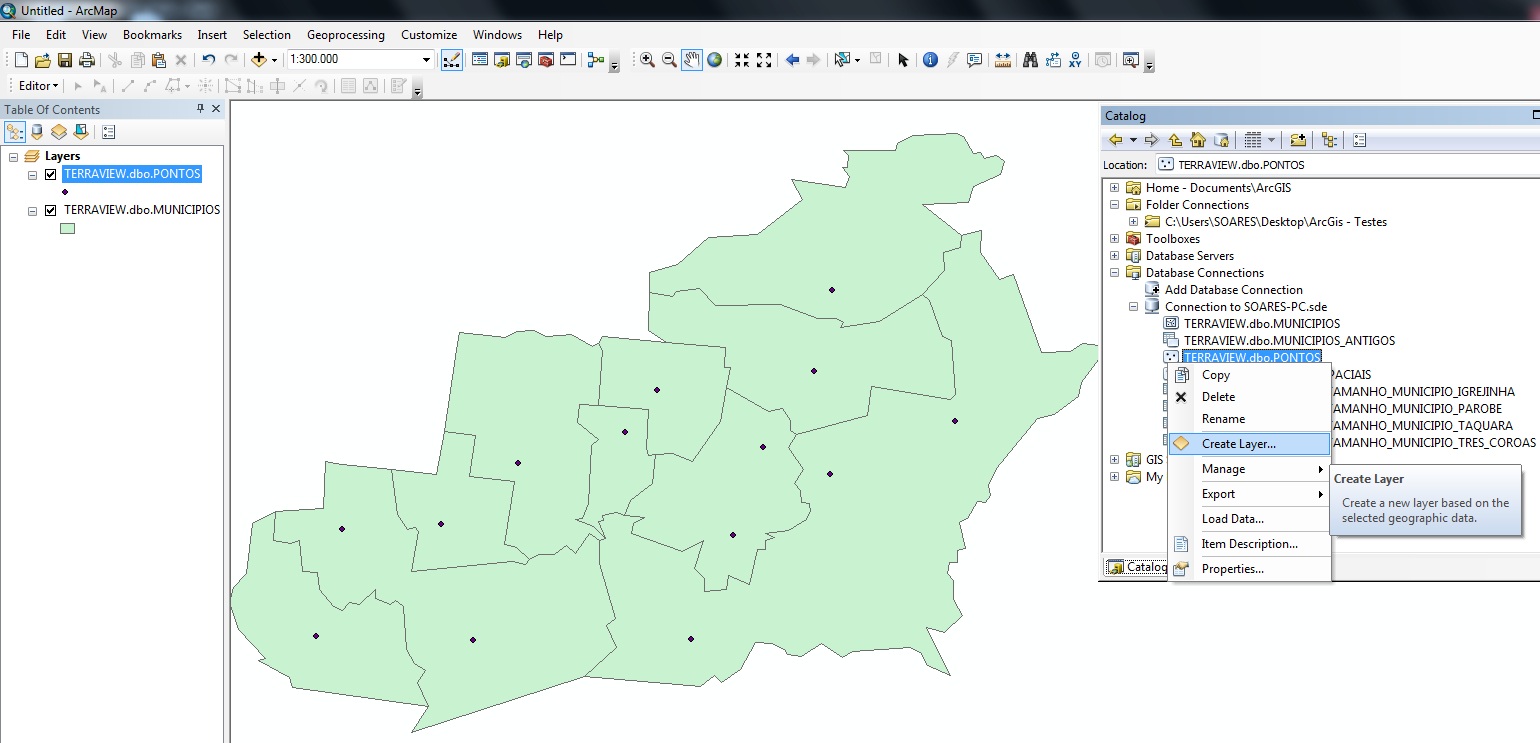


Figura 47 – Adicionando uma nova *Layer*.

Como na tabela MUNICIPIOS já existem as informações de área e perímetro, as primeiras análises no ArcMap foram, para fins de comparação, para descobrir estas mesmas informações.

Para a realização do cálculo da área e perímetro foi necessária a exibição da tabela MUNICIPIOS e então, como exibe a figura 48, a opção “*Calculate Geometry*” (Calcular a Geometria) foi selecionada.

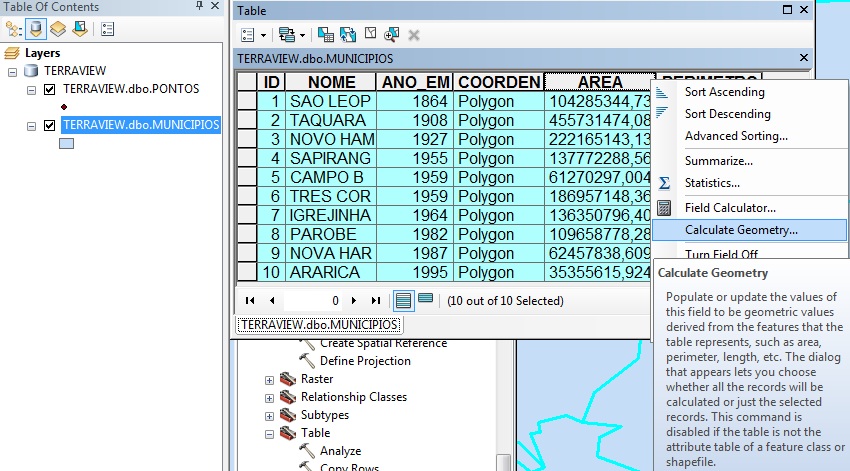


Figura 48 – Opção de calcular a geometria.

Após selecionar a opção “*Calculate Geometry*” (Calcular a Geometria), para calcular a área dos polígonos, foi necessária a configuração do sistema de coordenadas (*SRID* 4326 do SQL Server 2008) e a unidade de medida (metros quadrados) a ser utilizada. Para que fosse possível realizar o cálculo do perímetro, apenas a opção *Property* (Propriedade) foi alterada para *Perimeter* (Perímetro). A figura 49 demonstra a configuração utilizada para o cálculo da área.

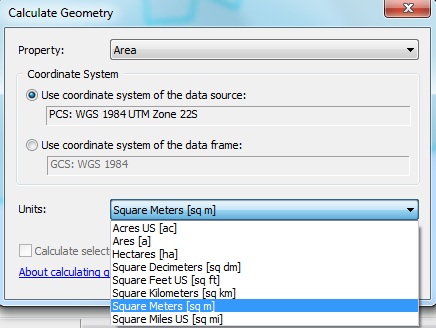


Figura 49 – Configuração de cálculo da área ou perímetro.

O resultado dos cálculos da área e perímetro, demonstrados na figura 50, foram realizados com sucesso. Porém foi possível identificar que tanto o cálculo da área quanto o do perímetro, realizados no ArcMap, forneceram resultados diferentes dos encontrados nas análises realizadas no SQL Server 2008. Conforme a figura 50, os resultados do ArcMap (em verde) foram, nos dois casos, inferiores aos encontrados no SQL Server 2008 (em branco).

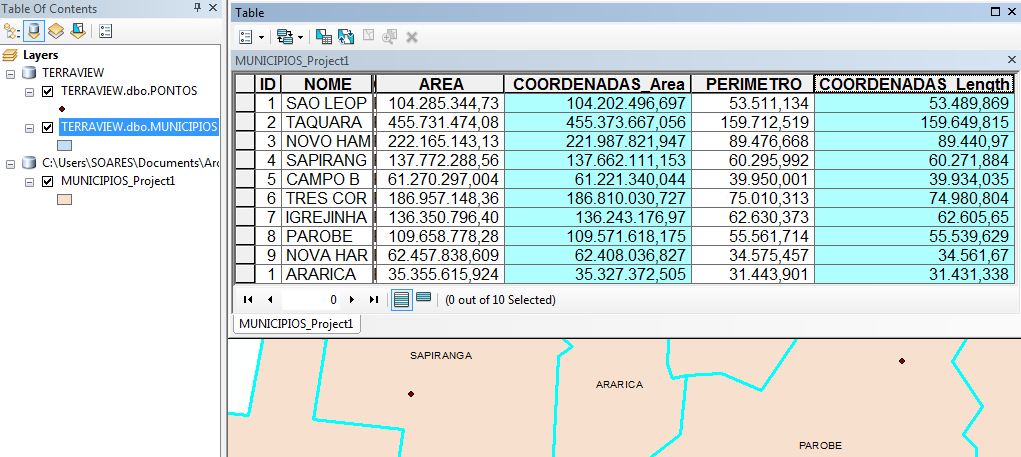


Figura 50 – Diferenças entre os resultados de área e perímetro.

Para que fosse realizado o cálculo da distância entre dois pontos, foi utilizada a função *Measure* (Medida). A figura 51 demonstra o cálculo da distância entre a prefeitura do município de Taquara e seu atual distrito Fazenda Fialho. O resultado encontrado na ferramenta ArcMap foi exatamente igual ao encontrado no SQL Server 2008.

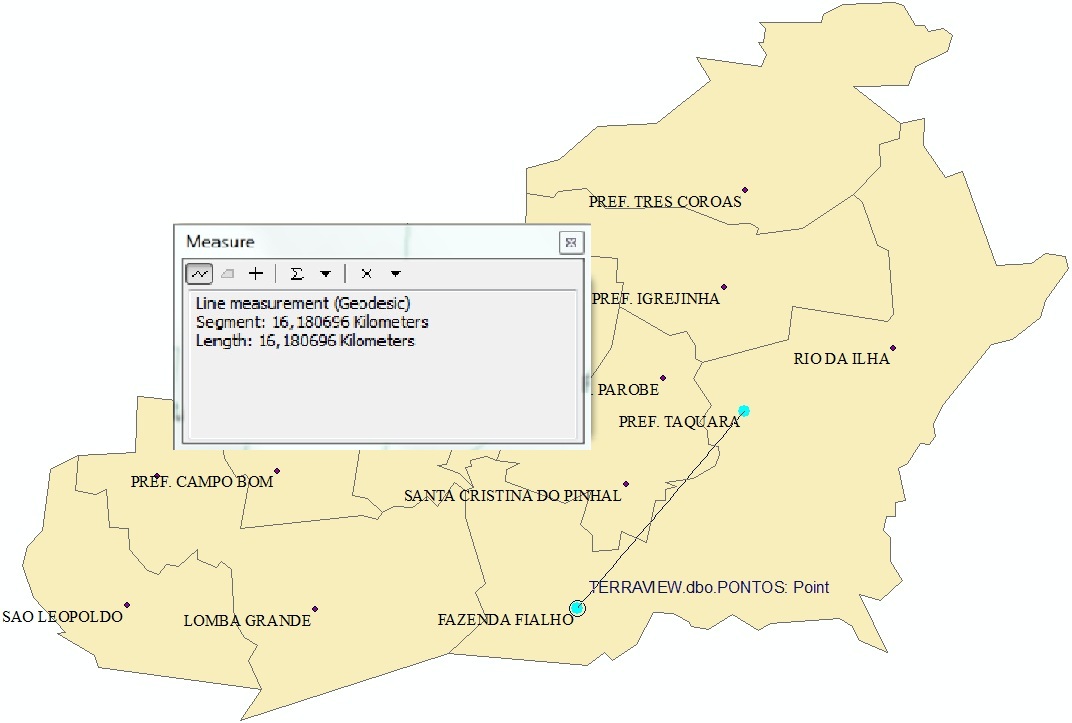


Figura 51 – Distância entre dois pontos.

O ArcMap, assim como o SQL Server 2008, também dispões de uma função para calcular um *Buffer* de atuação. Como demonstrado anteriormente no SQL Server 2008, um *Buffer* foi criado nas prefeituras dos municípios de Novo Hamburgo e Taquara e também em seus respectivos distritos de Lomba Grande, Rio da Ilha e Fazenda Fialho. Na figura 52 é exibido o resultado da operação de *Buffer* em um raio de atuação de cinco quilômetros.

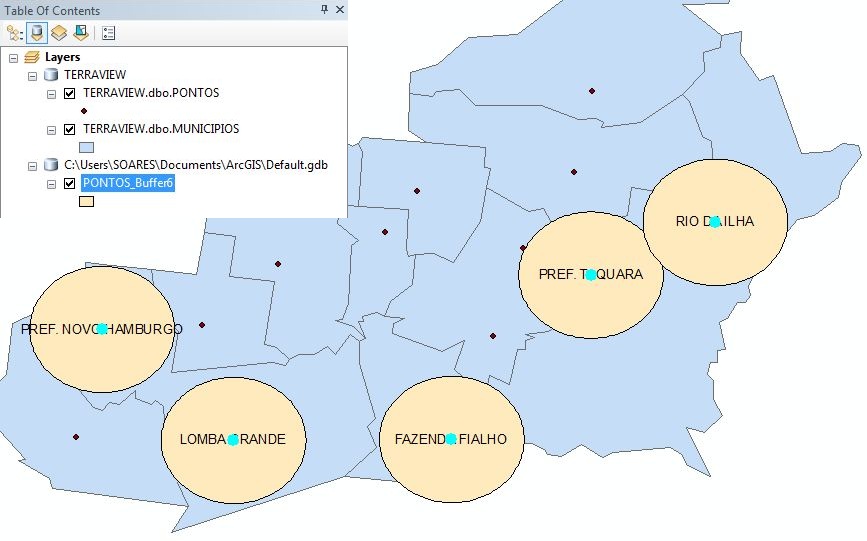


Figura 52 – *Buffer* com raio de cinco quilômetros.

Existe ainda a função *Dissolve* (Dissolver) que realiza a mesma operação da função *STUnion()* do SQL Server 2008. Conforme a figura 53, os municípios de Taquara, Parobé, Igrejinha e Três Coroas foram selecionados e a função *Dissolve* aplicada.

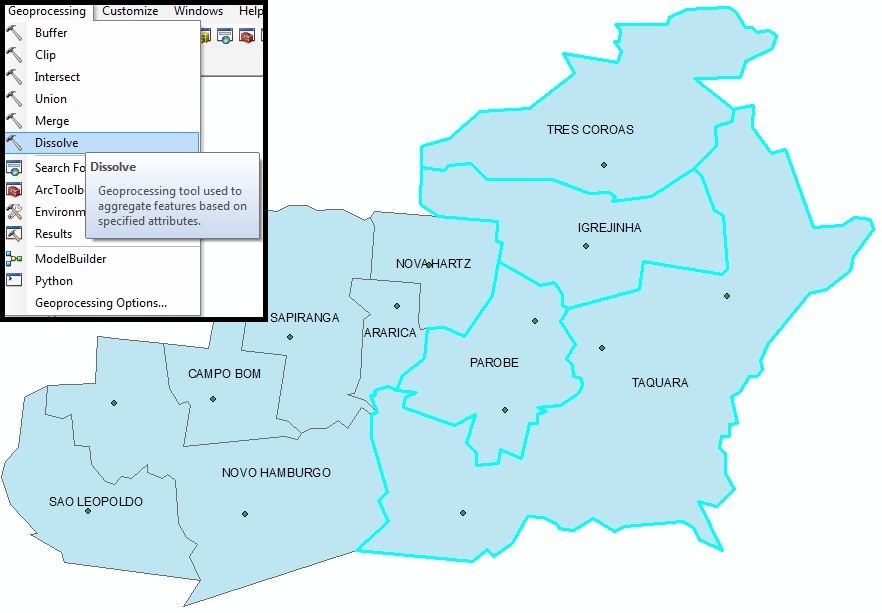


Figura 53 – Aplicação da função *Dissolve*.

Na figura 54 é exibido o resultado da função *Dissolve* que ao reunir os quatro municípios, formou o antigo município de Taquara, de acordo com o ano de 1908.



Figura 54 – Resultado da função *Dissolve*.

Com o desenvolvimento dos bancos de dados geográficos, as ferramentas SIG se tornaram muito mais robustas. Ao se utilizar o SGBD SQL Server 2008 integrado a ferramenta SIG ArcGIS 10.1, tornou-se possível que os dados manipulados no ArcGIS fossem armazenados de forma adequada em um banco de dados geográfico. Com esta divisão de tarefas, a ferramenta SIG fica encarregada somente de manipular os dados e o SGBD geográfico se encarregada de armazenar os dados conforme cada situação, permitindo com isto uma transferência de dados muito mais veloz entre as duas ferramentas.

# CONCLUSÃO

Devido ao fato de empresas estarem sempre em desenvolvimento de novos softwares, surgiu a necessidade de se criarem padrões de desenvolvimento onde estes novos softwares pudessem se comunicar entre si e com isso a interoperabilidade pudesse ser garantida.

Para garantir a interoperabilidade entre softwares de diferentes empresas, foi criado o *Open Geospatial Consortium* que através de seus padrões de desenvolvimento melhoraram a comunicação e o compartilhamento de dados entre softwares de diferentes empresas. Sem a criação destes padrões, a possibilidade de que cada novo software desenvolvido tivesse seus próprios padrões seria muito grande. Além disso, com a criação do OpenGIS, foi aberto um mercado que antes se encontrava dominado por ferramentas proprietárias e de alto custo.

A partir da utilização do geoprocessamento, aliado ao seu conjunto de ferramentas, como por exemplo, o SIG, uma ferramenta muito útil em nossos dias, é possível realizar as mais distintas tarefas em diferentes áreas de atuação, como por exemplo, a construção de residências e viadutos em áreas como a da arquitetura, topografia e engenharia.

Após os SIGs terem sido criados, surgiu a necessidade de se armazenar seus dados de uma melhor forma. A criação de bancos de dados geográficos foi muito importante, pois com os mesmos os SIGs puderam realizar tarefas mais complexas de forma rápida e com um apropriado armazenamento dos dados. Além disso, os bancos de dados geográficos são de grande utilidade para o ser humano, pois sem a utilização dos mesmos, muitas das tarefas de profissionais da área geográfica seriam mais complexas, ou impossíveis de serem resolvidas.

Através dos estudos realizados, foi possível concluir que todos os três sistemas gerenciadores de bancos de dados estudados são, na teoria, muito poderosos em seus módulos espaciais. Ambos tiveram seu desenvolvimento baseado nas especificações da OGC, garantindo com isso a interoperabilidade dos dados.

Mesmo se tratando de um SGBD ainda iniciante no tratamento de dados espaciais, o SQL Server 2008 mostrou que sua experiência anterior no tratamento de dados não espaciais foi de grande serventia, pois com isto, conseguiu aliar o ótimo desempenho em bancos de dados com um módulo espacial bastante completo, cheio de recursos e suficientemente eficaz. Além disso, o SQL Server 2008 respondeu positivamente a todas as consultas realizadas, cumprindo com as informações contidas em sua documentação oficial. Uma vasta gama de ferramentas espaciais, 100% funcionais, para serem utilizadas pelo usuário. Possui um recurso de carga de dados, através de interface gráfica, de fácil utilização e muito intuitivo, facilitando assim as operações de importação e exportação dos dados.

O SIG ArcGIS, através de sua ferramentas extras, demonstrou ser uma excelente ferramenta para a manipulação de dados espaciais. Possui inúmeras funções e funcionalidades, permitindo ao usuário realizar operações das mais variadas formas. Em contraponto ao SQL Server 2008, o ArcGIS mostrou-se bastante intuitivo, apesar de não ser uma ferramenta de fácil utilização, necessitando de conhecimento básico sobre dados espaciais por parte do usuário. Mesmo assim, o fato de o ArcGIS não necessitar de conhecimento em programação e ser utilizado através de interface gráfica facilitou o seu uso.

Com a execução do presente trabalho, o pesquisador e demais interessados no assunto, mesmo sem muitos conhecimentos em SGBDs ou dados espaciais, ganham a possibilidade de ampliar seus conhecimentos acerca de todas as ferramentas apresentadas e com isto, novas formas ao tratamento de dados espaciais podem ser analisadas de maneira a enriquecer suas futuras experiências na área geográfica. Isto se dá pelo fato de que com esta pesquisa será possível conhecer, analisar e até resolver problemas de pesquisadores passados, atuais ou futuros que talvez tenham tido suas pesquisas pausadas por não encontrarem materiais suficientes para auxiliá-los.

# REFERÊNCIAS BIBLIOGRÁFICAS

AITCHISON, Alastair. **Beginning Spatia with SQL Server 2008**. Apress, New York, 2009. 458 p.

BORGES, Karla A. V. et al. **Banco de Dados Geográficos**. São José dos Campos: INPE - São Paulo, 2006.

CÂMARA, Gilberto et al. **Banco de Dados Geográficos**. São Paulo: MundoGEO, 2005. 506 p.

CARRILLO, German. **Consola SQL para el Plugin de pgAdmin: PostGIS Viewer**. 2012. Disponível em: <

http://geotux.tuxfamily.org/index.php/es/component/k2/item/293-consola-sql-para-plugin-pgadmin-postgis-viewer>. Acesso em: 12/12/2012.

CODEPLEX. **SQL Server Spatial Tools**. 2010. Disponível em: <sqlspatialtools.codeplex.com>. Acesso em: 06/10/2012.

ESRI. **ArcGIS**. 2012. Disponível em: <www.esri.com/software/arcgis>. Acesso em: 20/10/2012

ESRI. **Esri Shapefile Technical Description**. 1998. Environmental Systems Research Institute, Inc., Redlands. 1998. 34 p.

IBGE, ed. (2010). **Instituto Brasileiro de Geografia e Estatística**. Disponível em: <http://www.ibge.gov.br/home/>. Acesso em: 07/10/2012.

LOURENÇO, Paula Moreira Barbosa. **Um estudo sobre recursos de tratamento de dados espaciais em SGBDs Geográficos.** XI Curso de Especialização em Geoprocessamento, Belo Horizonte – BH, 2008. 71 p.

MALCOLM, Graeme. **Microsoft SQL Server 2008 - Fornecendo Inteligência de Localização com Dados Espaciais**. Microsoft Corporation, 2007. Disponível em: <http://brasil.microsoft.com.br>. Acesso em: 09/06/2012. 12 p.

MICROSOFT. **Manuais Online do SQL Server 2008**. Microsoft Corporation, 2008. Disponível em <http://msdn.microsoft.com/pt-br/library/bb500434.aspx>. Acesso em: 05/04/2012.

MURRAY, Chuck. **Oracle® Spatial User’s Guide and Reference 10g Release 1(10.1)**. Oracle Corporation, Redwood City, 2003. 602 p.

OGC, ed. (2012). **OPEN GEOSPATIAL CONSORTIUM**. Disponível em: <http://www.opengeospatial.org>. Acesso em: 19/04/2012.

ORACLE, Spatial. **User´s Guide and Reference 10g**. Oracle, 2003.

PAIVA, Vivian Maria. **Banco de Dados Geográficos: Estudo de Caso da Aplicação das Extensões**. Dissertação de Monografia - UFJF. Julho – 2007

PETEK, Matic. **GIS Extension for Oracle SQL Developer**. 2006. Disponível em < https://sites.google.com/site/mygeoraptor/>. Acesso em: 12/12/2012.

POSTGIS, Manual. **PostGIS 2.0.0 Manual**. Refractions Research. 2011. 591 p.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. Ed. Universidade Feevale, Novo Hamburgo – RS, 2009. 288 p.

QUEIROZ, Gilberto Ribeiro; Ferreira, Karine Rreis**. SGBD com extensões espaciais**. Curitiba – PR, Mundo GEO, 2005.

QUEIROZ, Gilberto Ribeiro; Ferreira, Karine Rreis**. Tutorial sobre Bancos de Dados Geográficos.** GeoBrasil, 2006.

SANTOS, André Milke dos. **Extensões Espaciais de Sistemas Gerenciadores de Banco de Dados**. Graduação – Universidade Feevale, Novo Hamburgo – RS, 2006. 99 p.

SILVA, Reginaldo Macedônio da. **Introdução ao Geoprocessamento: conceitos, técnicas e aplicações**. 2. Ed. Universidade Feevale, Novo Hamburgo - RS, 2010. 184 p.

THOMÉ, R. **Interoperabilidade em Geoprocessamento: Conversão entre Modelos Conceituais de Sistemas de Informação Geográfica e Comparação com o Padrão**. OpenGIS. (INPE-7266-TDI/708). Dissertação (Mestrado em Computação Aplicada). São José dos Campos: INPE, 1998, 196 p.

VINHAS, Lúbia. **Um Subsistema Extensível para o Armazenamento de Geo-Campos em Bancos de Dados Geográficos**. 114 f. Tese (Doutorado) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos – SP, 2006. 114 p.

YUAN, May. **Development of a Global Conceptual Schema for Interoperable Geographic Information**. Disponível em: <http://www.ncgia.ucsb.edu/conf/interp97/program/papers/yuan/yuan.html>. Acesso em: 28/05/2012.