

Spatial Database Systems

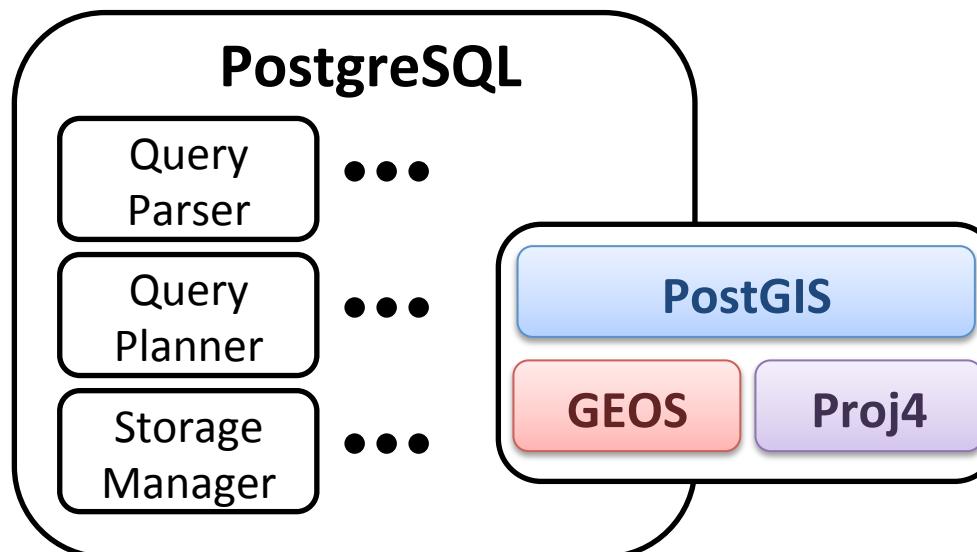
PostGIS



Gilberto Ribeiro de Queiroz
[\(gribeiro@dpi.inpe.br\)](mailto:gribeiro@dpi.inpe.br)

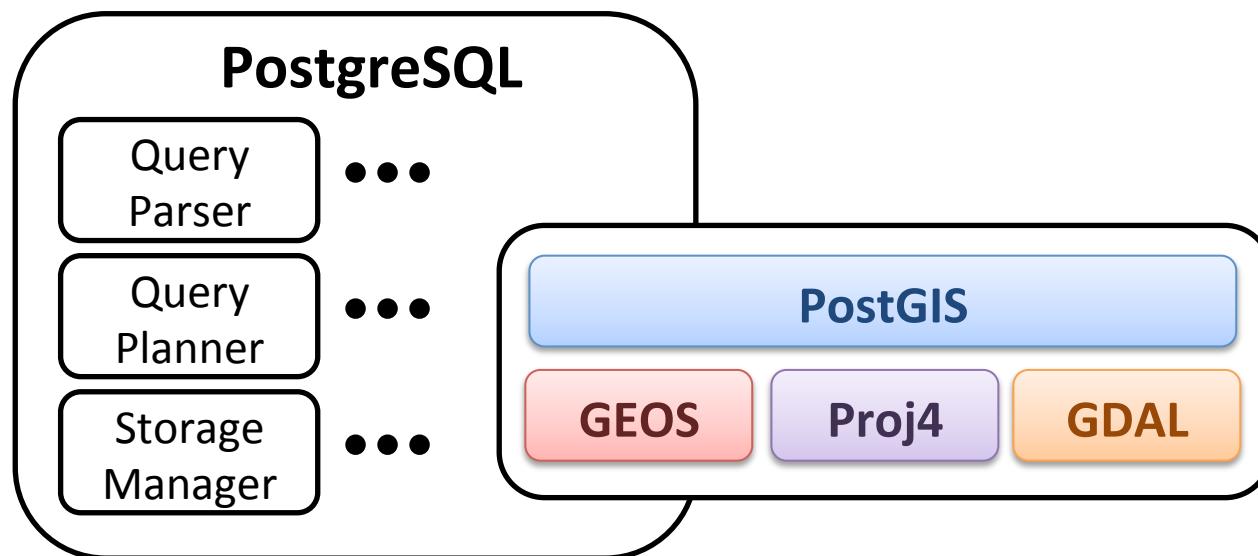
PostGIS

- Projeto de software livre (GPLv2) desenvolvido inicialmente pela empresa Canadense Refractions Research:
<http://postgis.refractions.net/>
- Extensão geográfica para o SGBD-OR PostgreSQL:
 - Inicialmente:
 - tipos geométricos e operadores espaciais OGC SFS.
 - Índice espacial: árvore-R sobre GiST.



PostGIS

- Grande evolução nos últimos 3 anos:
 - Tipos circulares e compostos
 - Tipo geográfico
 - Raster
 - Topologia
 - Redes
 - Geocodificação de endereços



Vamos criar um novo banco de dados
com a extensão PostGIS habilitada

PostGIS Versão 2.0

Se usarmos o template1 para criação do banco

```
psql -U postgres -d postgres
```

```
CREATE DATABASE bdcurso TEMPLATE template1;
```

- Conectar no novo banco de dados:
\c bdcurso

```
CREATE EXTENSION postgis;
```

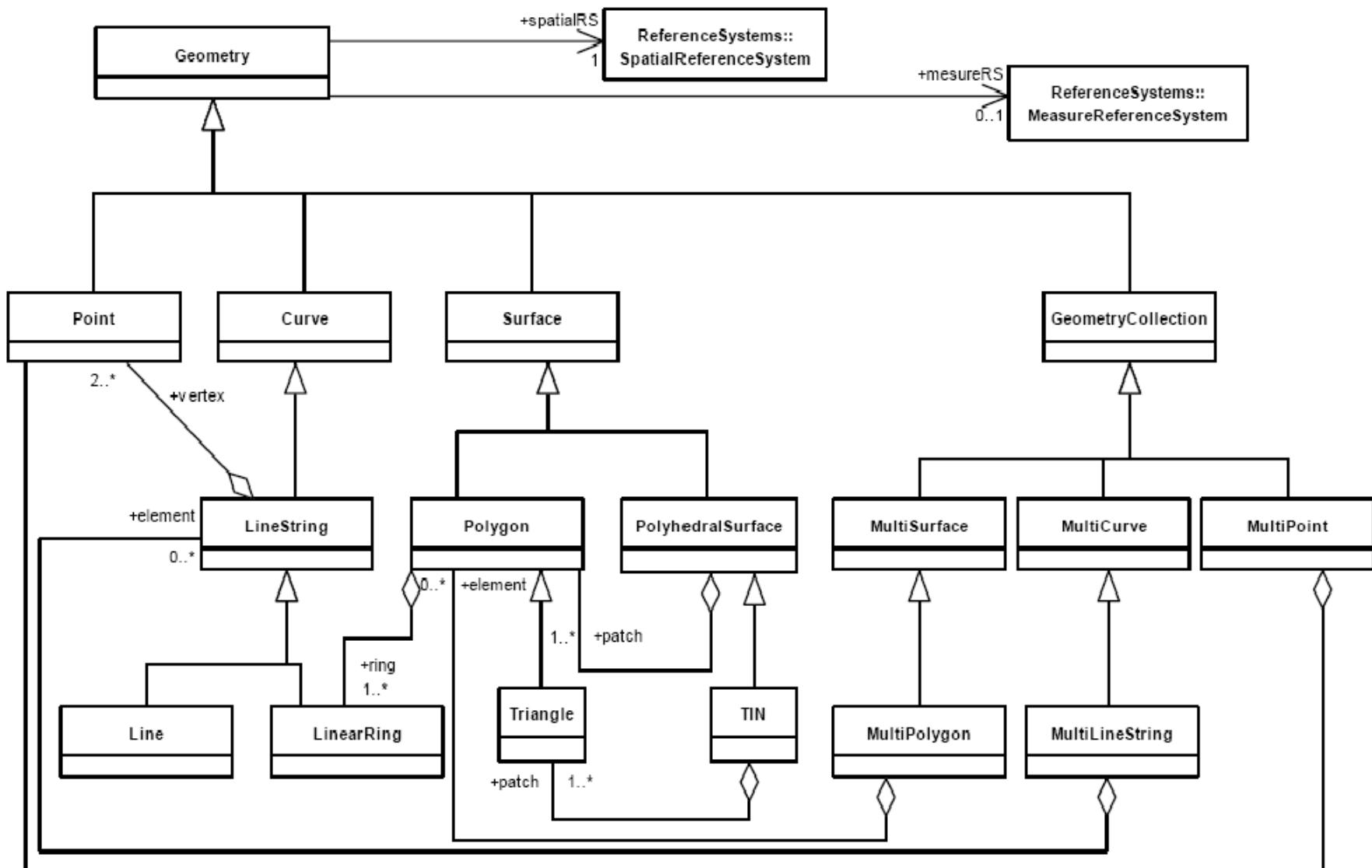
- Verificando a versão do PostGIS:
SELECT postgis_full_version();

Nota: O PostGIS necessita da extensão PL/pgSQL

O que é a especificação da Simple Feature (SFS)?

- Especificação criada pelo consórcio OGC (Open Geospatial Consortium) que trata das questões de representação da componente espacial vetorial de dados geográficos:
 - Aspectos relativos à representação de pontos, linhas e polígonos.
- O OGC é um consórcio formado por empresas, universidades e agências governamentais de diversos países.
- Um de seus objetivos é promover o desenvolvimento de padrões que facilitem a interoperabilidade entre sistemas de informação geoespaciais.
- Parte do trabalho do OGC é apresentado sob a forma de especificações abertas de interfaces e padrões de intercâmbio.
- Site: <http://www.opengeospatial.org/>

SFS: Modelo Geométrico



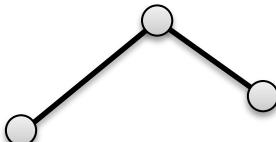
Fonte: OGC (2012a)

Alguns tipos geométricos do PostGIS

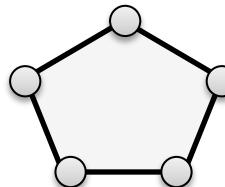
Point



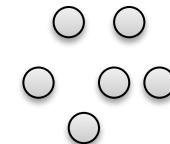
LineString



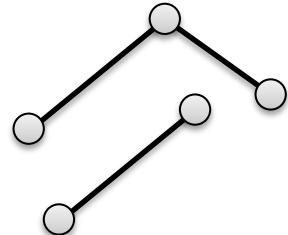
Polygon



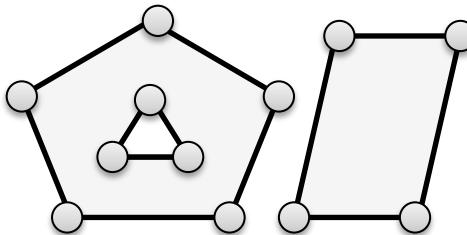
MultiPoint



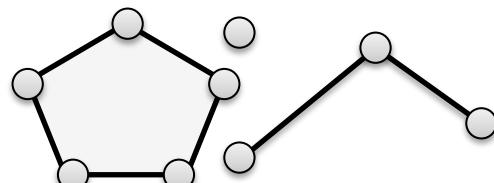
MultiLineString



MultiPolygon



Geometry Collection



```
CREATE TYPE Geometry
(
internallength = variable,
input = geometry_in,
output = geometry_out,
send = geometry_send,
receive = geometry_recv,
typmod_in = geometry_typmod_in,
typmod_out = geometry_typmod_out,
delimiter = ':',
alignment = double,
analyze = geometry_analyze,
storage = main);
```

Criando uma tabela com coluna geométrica

```
CREATE TABLE estacoes_pluviometricas
(
    gid INT4,
    location GEOMETRY(POINT, 4618),
    nome VARCHAR(25)
);
```

```
INSERT INTO estacoes_pluviometricas
VALUES(1, ST_GeomFromText('POINT(-46.98 -19.57)', 4618),
       'DINIZ-ARAXA');
```

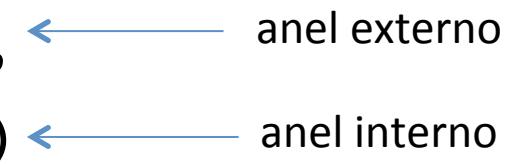
```
INSERT INTO estacoes_pluviometricas
VALUES(2, ST_GeomFromText('POINT(-43.59 -20.37)', 4618),
       'QUEIROZ-OURO-PRETO');
```

OGC SFS: Well-Known Text (WKT)

POINT(0 0)

LINESTRING(0 0, 1 1, 1 2)

POLYGON((0 0, 4 0, 4 4, 0 4, 0 0),
(1 1, 2 1, 2 2, 1 2, 1 1))



MULTIPOINT(0 0, 1 2, 1 3, 1 4, 2 2, 3 3)

MULTILINESTRING((0 0, 1 1, 1 2),
(2 3, 3 2, 5 4))



Fonte: Adaptado do Manual PostGIS 2.0

Well-Known Text (WKT)

```
MULTIPOLYGON(((0 0, 4 0, 4 4, 0 4, 0 0),  
                (1 1, 2 1, 2 2, 1 2, 1 1)),  
                ((5 0, 5 4, 6 4, 5 0)))
```

1^a Polígono
2^a Polígono

```
GEOMETRYCOLLECTION(  
    POLYGON((0 0, 4 0, 4 4, 0 4, 0 0),  
            (1 1, 2 1, 2 2, 1 2, 1 1)),  
    MULTIPOLYGON((0 0, 1 2, 1 3, 1 4, 2 2, 3 3)),  
    MULTILINESTRING((0 0, 1 1, 1 2),  
                    (2 3, 3 2, 5 4)))  
)
```

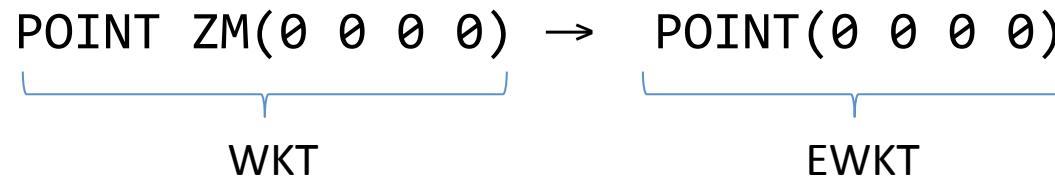
Fonte: Adaptado do Manual PostGIS 2.0

PostGIS EWKT

- O formato WKT definido pelo OGC não possibilita informar o sistema de referência espacial da geometria. O EWKT do PostGIS possibilita:

SRID=32632;POINT(0 0)

- Existem notações diferentes para geometrias com Z e M:

POINT ZM(0 0 0 0) → POINT(0 0 0 0)


Recuperando linhas com geometrias

```
SELECT gid, location  
FROM estacoes_pluviometricas
```

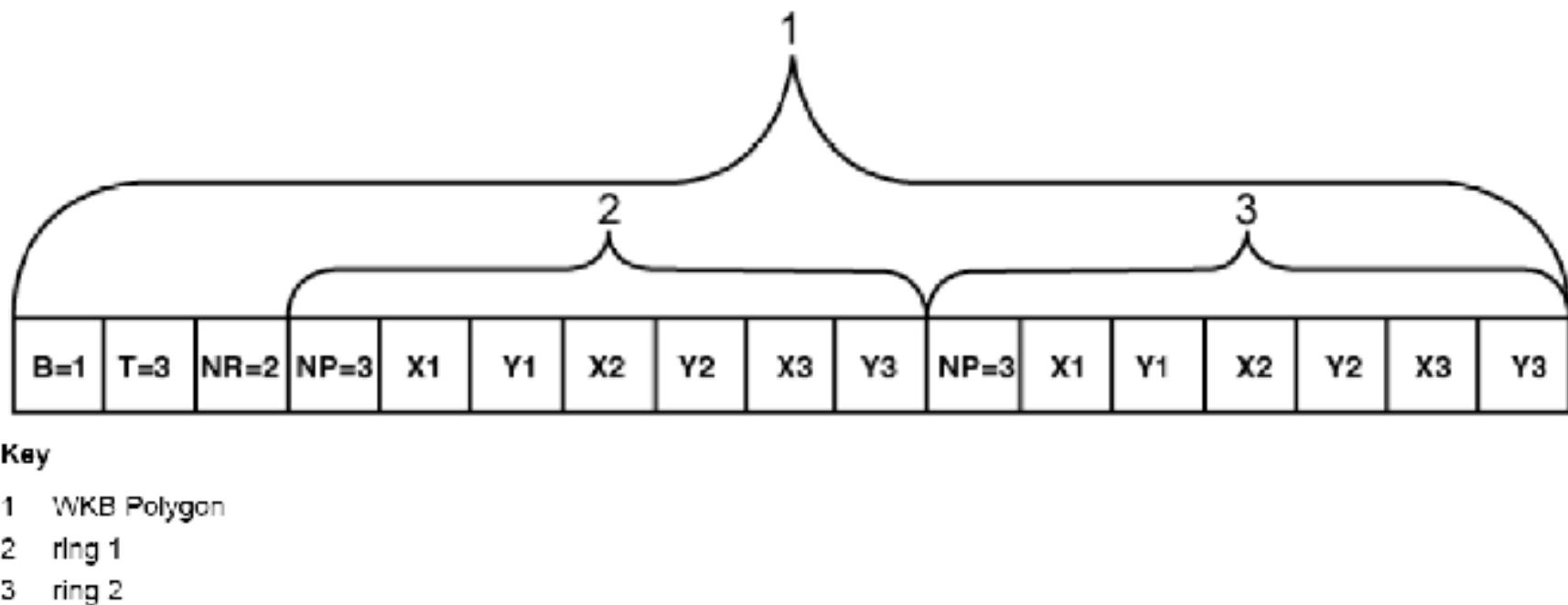
```
SELECT gid, ST_AsText(location)  
FROM estacoes_pluviometricas
```

- Quando usamos a notação EWKT:

```
ST_GeomFromEWKT  
ST_AsEWKT
```

Well-Known Binary (WKB)

- Versão binária para representação das geometrias.



**Figure 25: Well-known Binary Representation for a geometric object
in NDR format ($B = 1$)
of type Polygon ($T = 3$)
with 2 LinearRings ($NR = 2$)
each LinearRing having 3 points ($NP = 3$)**

Fonte: OGC (2012)

PostGIS

- TRIANGLE, TIN
- 2D, 3D, 2D + M, 3D + M
- Interpolação linear e circular
- Tipos circulares e compostos (SQL-MM Spatial):
 - CircularString
 - CompoundCurve
 - CurvePolygon
 - MultiCurve
 - MultiSurface

Nota: SQL/MM

- Como o armazenamento de dados espaciais passou a desempenhar um papel importante na indústria, a especificação OGC passou a ser considerada pela ISO, que criou a especificação SQL/MM Spatial:
 - Os tipos geométricos e operações passam a ter o prefixo “ST_”:
 - Ex: ST_Polygon, ST_point, ST_Area, ST_centroid.

Em versões anteriores ao PostGIS 2.0

- Forma correta de criar uma Tabela Espacial:

Primeiro Passo:

```
CREATE TABLE estacoes_pluviometricas
(
    gid INT4,
    nome VARCHAR(25)
);
```

Segundo Passo:

```
SELECT AddGeometryColumn('public',
                         'estacoes_pluviometricas',
                         'location',
                         4618, 'POINT', 2)
```

Metadado Colunas Tipos Geométricos

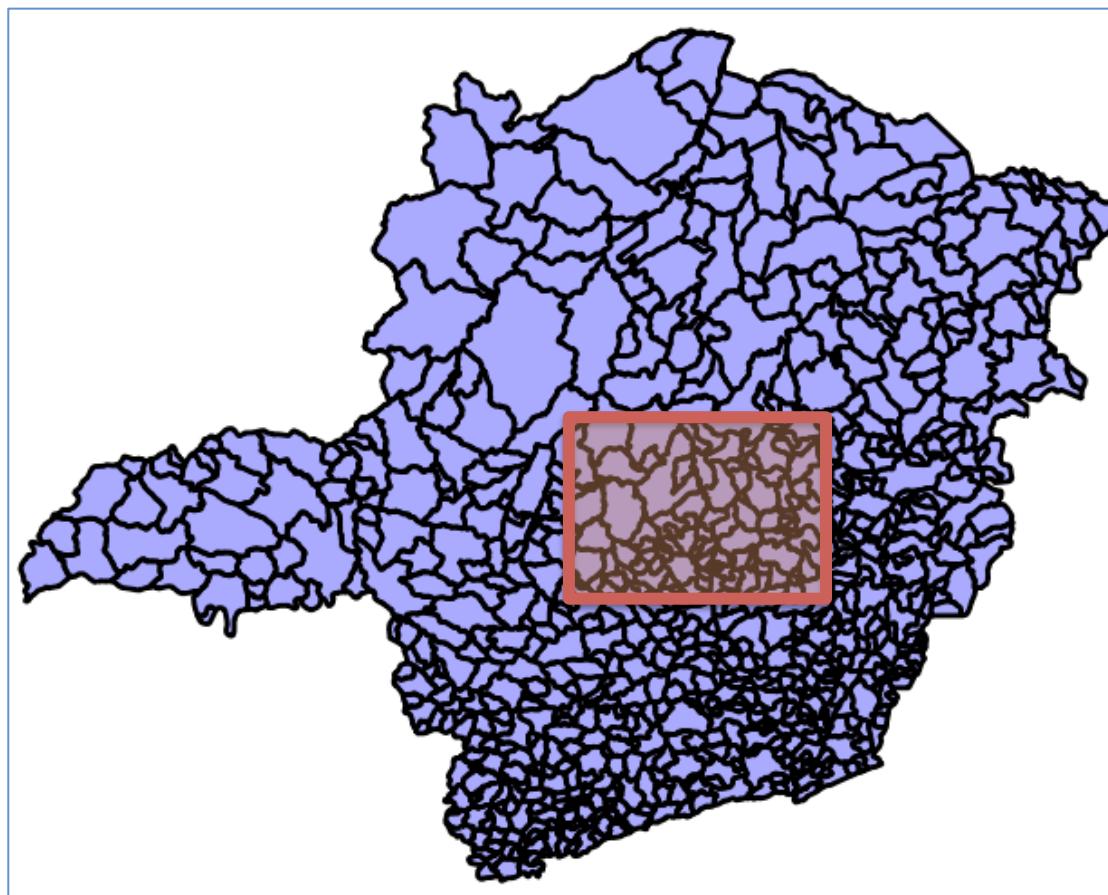
- Na versão 2.X do PostGIS:
 - **View:** geometry_columns
 - **Tabela:** spatial_ref_sys
- Na versão 1.X do PostGIS:
 - **Tabela:** geometry_columns
 - **Tabela:** spatial_ref_sys
- Vamos consultar as tabelas/views de metadado:

```
SELECT * FROM spatial_ref_sys WHERE srid = 4618
SELECT * FROM spatial_ref_sys WHERE srid = 31984
SELECT * FROM geometry_columns
WHERE f_table_name = 'estacoes_pluviometricas'
```

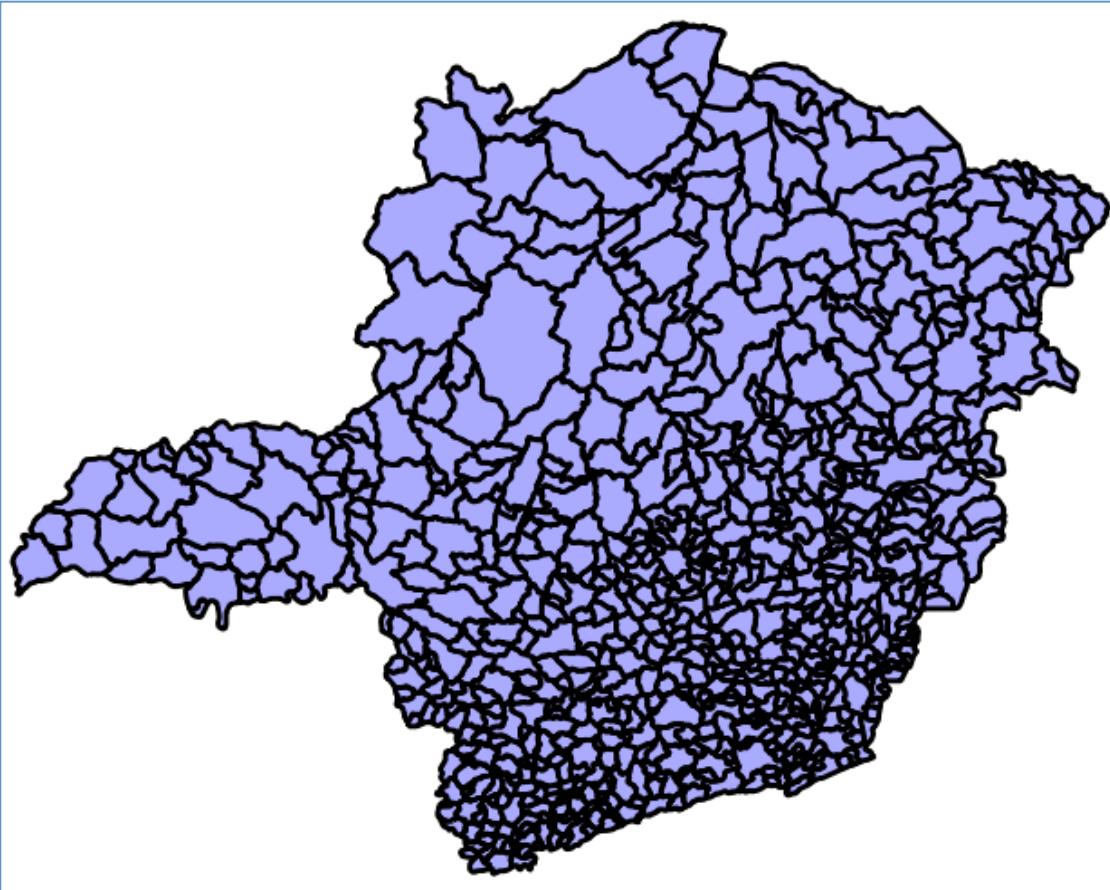
Quando trabalhos com dados geoespaciais,
nos referimos a alguns tipos de consultas com
nomes especiais

Consultas Espaciais: Intervalos no R^d

- Também conhecida como “range query” ou “region query”:
 - **Definição:** dado um retângulo (região de consulta), recuperar todos os elementos interceptam este dado retângulo;



Importar o dado de municípios de Minas Gerais



Arquivo:
dados/shp/mg_municipios

Tipo de dado:
Poligonos (756)

Sistema de Referência Espacial:
4618 => Lat/Long SAD/69

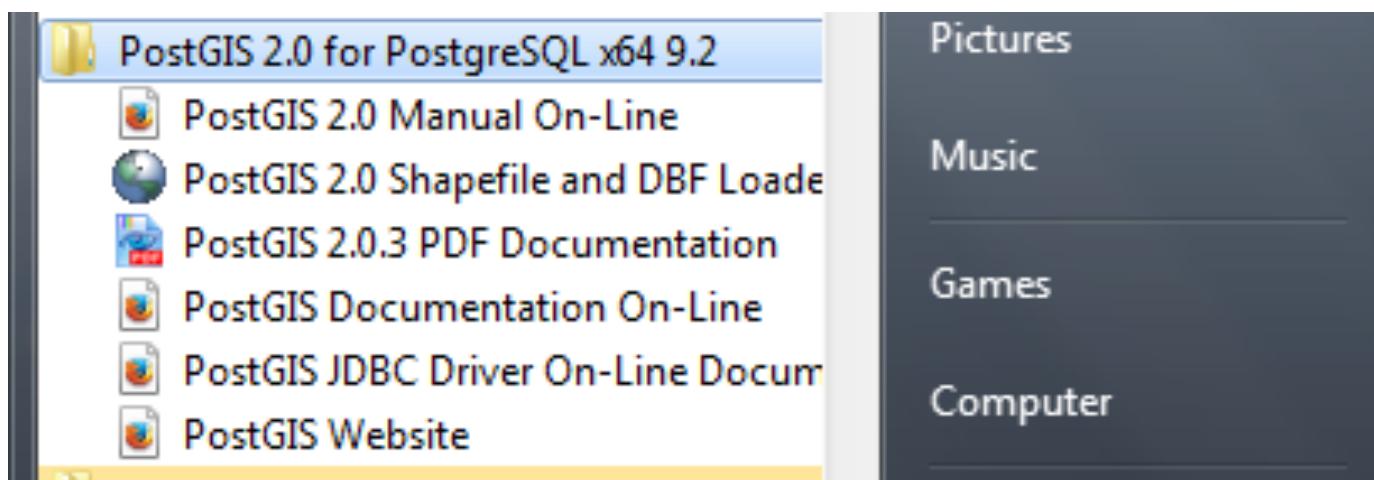
Nome da tabela a ser criada:
mg_municipios

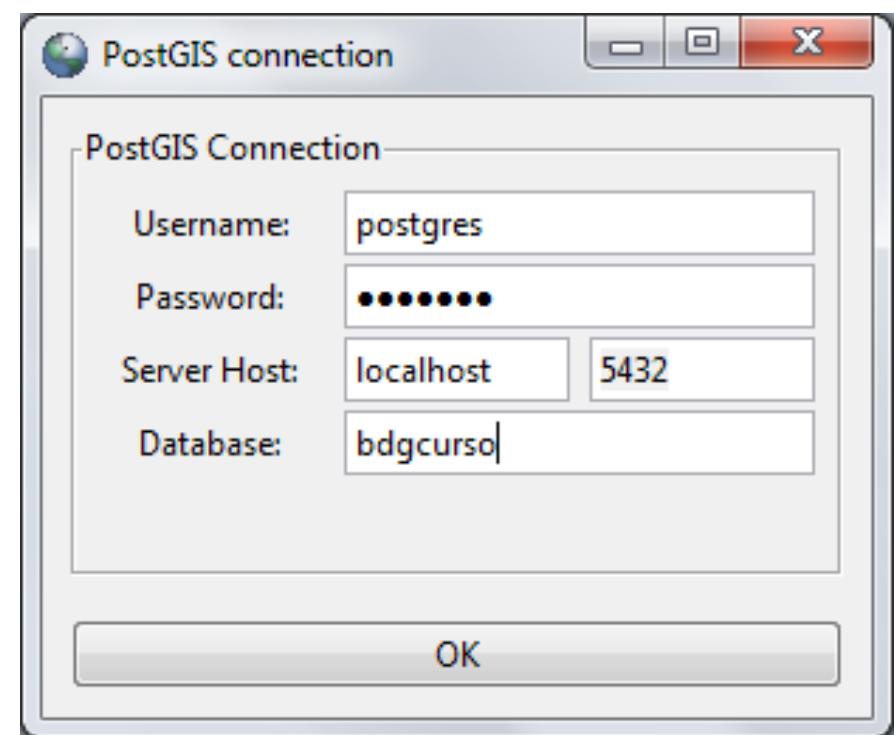
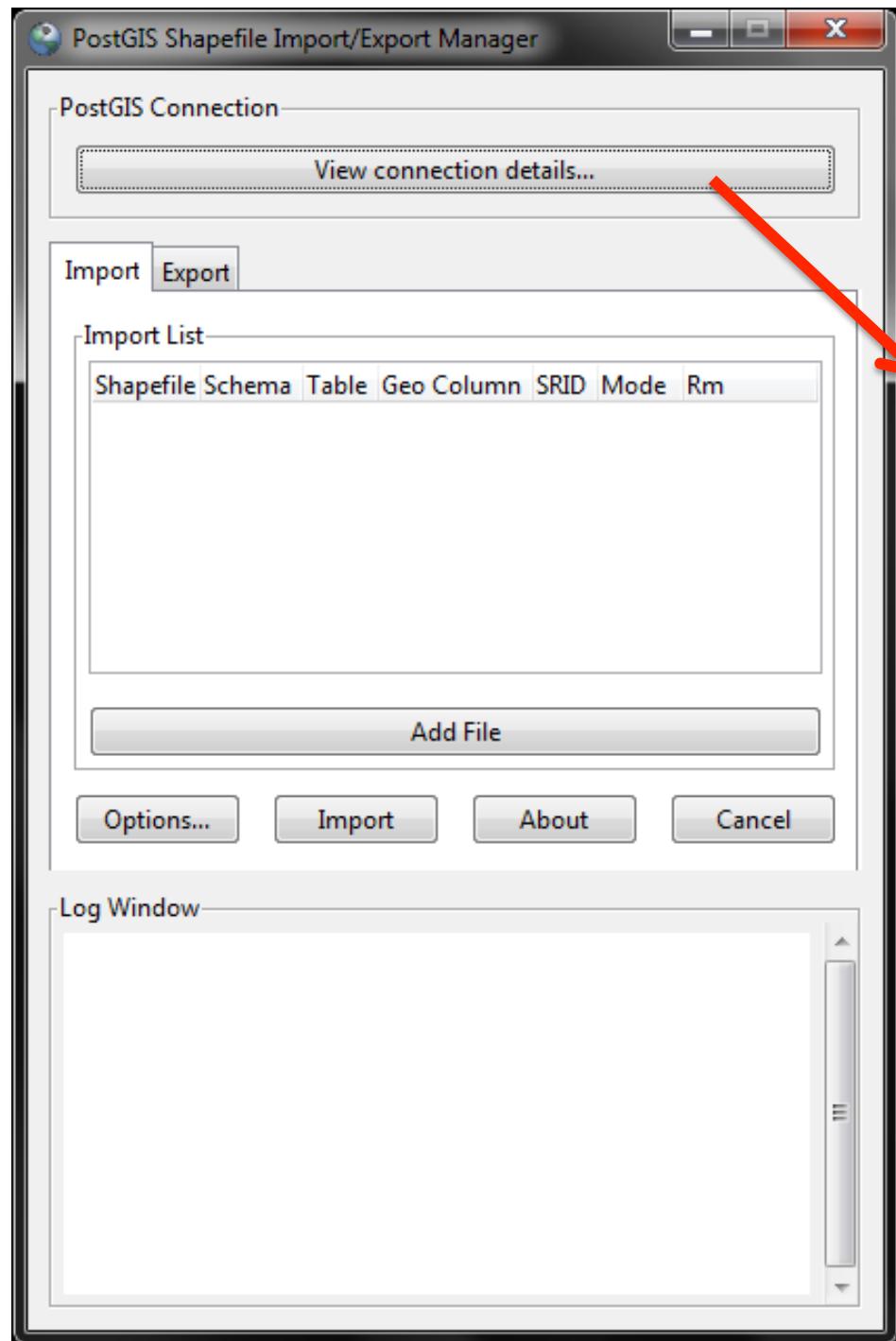
Codificação dos caracteres :
LATIN1

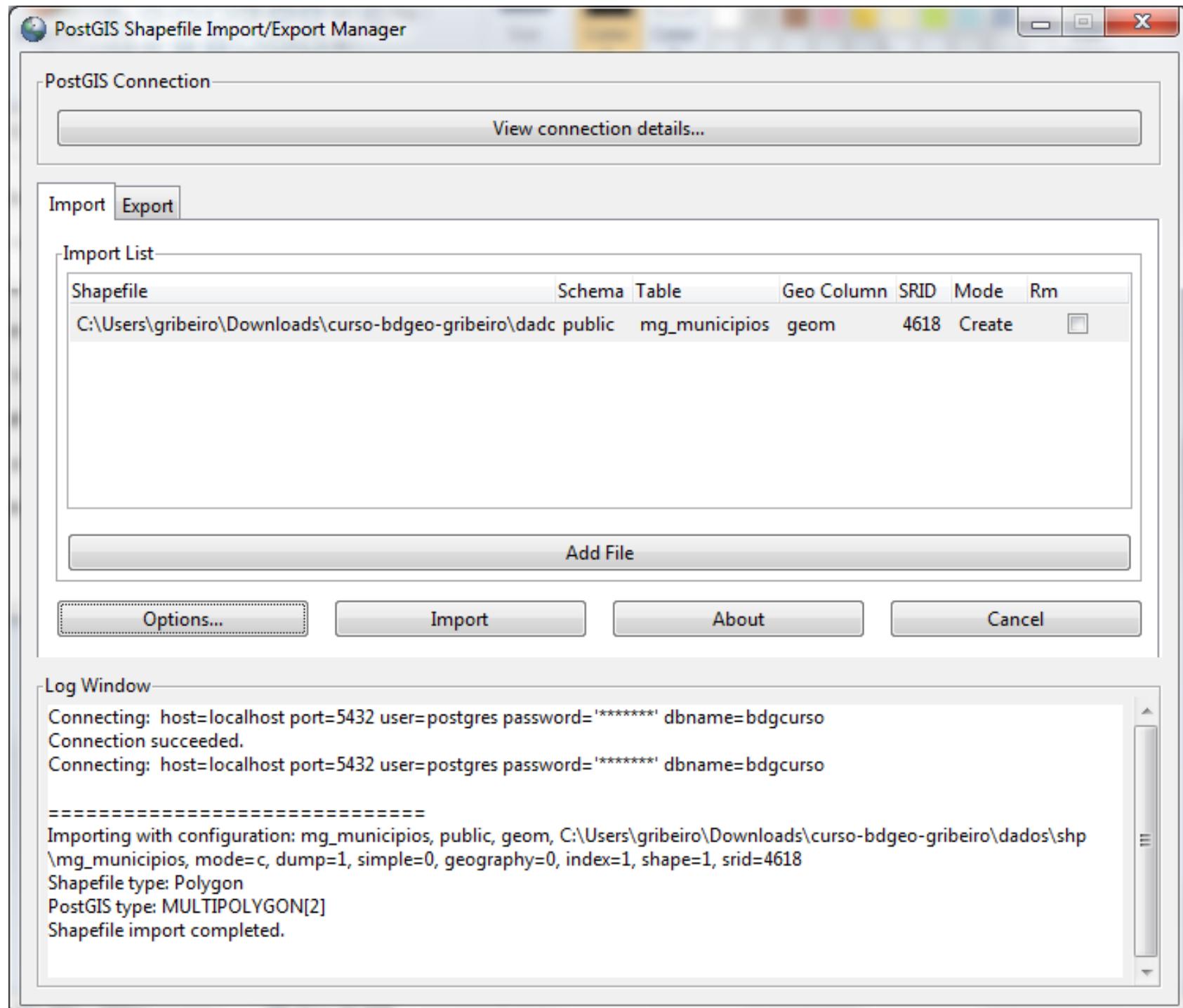
Shapefile → SQL PostgreSQL + PostGIS

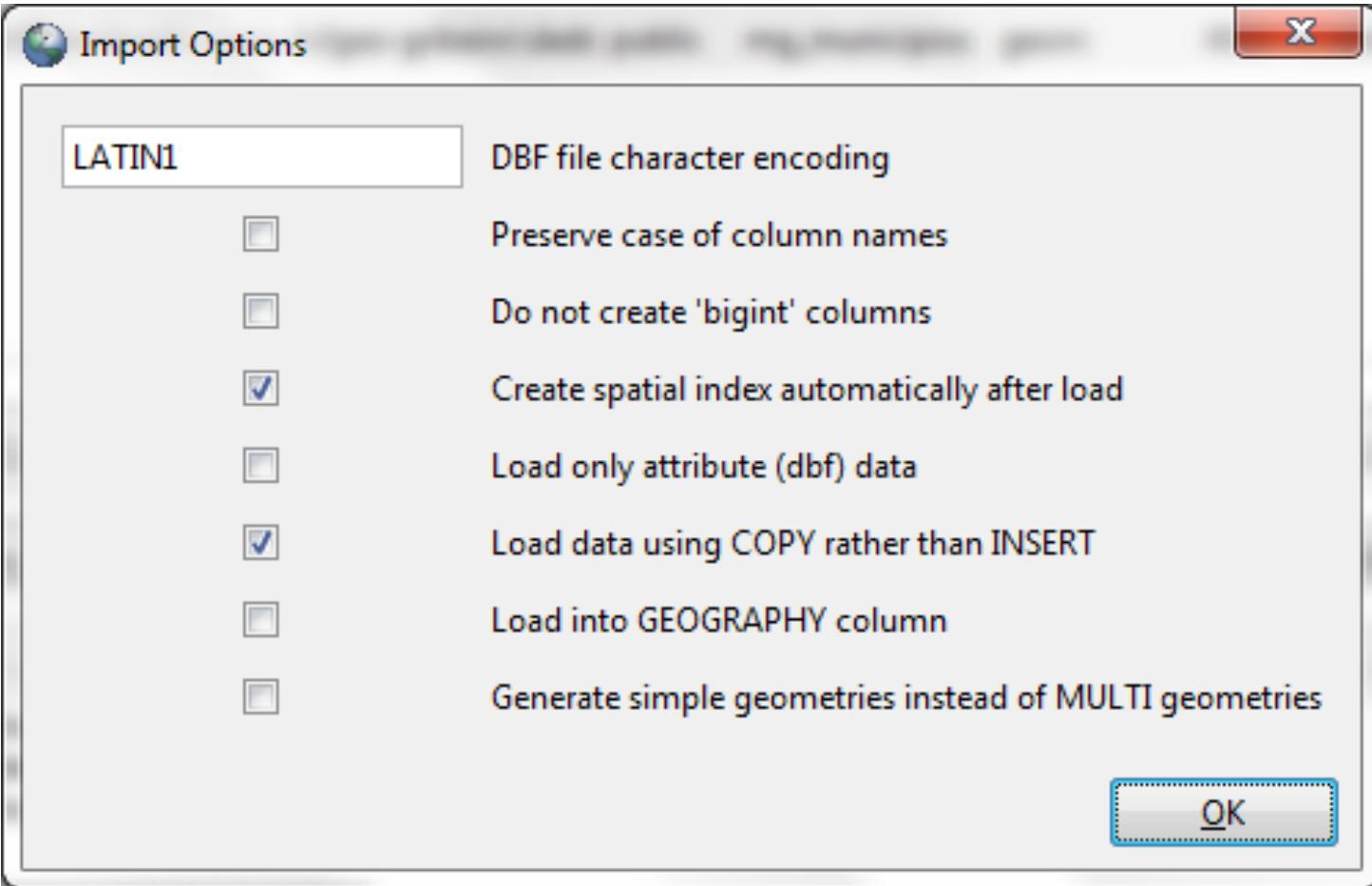
- Arquivo Shape:
 - .shp = contém a parte geométrica
 - .dbf = contém a parte alfa-numérica (string, number, date)
 - .shx = índice utilitário
 - .prj = sistema de referência espacial
- Tabelas PostgreSQL+PostGIS:
 - Colunas podem conter geometrias
 - Colunas podem conter atributos convencionais
- Em geral, um arquivo Shapefile irá corresponder a uma Tabela PostgreSQL + PostGIS

Abrir a interface gráfica de importação de Shapefiles









Outra opção seria usar o utilitário
shp2pgsql na linha de comando

```
shp2pgsql.exe -i -s 4618 -W LATIN1 mg_municipios.shp  
mg_municipios > mg_municipios.sql
```

Vamos visualizar nossa tabela no
Quantum GIS

Referências

- OGC. *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture.* Available at: <http://www.opengeospatial.org>. Access: October, 2012.
- OGC. *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option.* Available at: <http://www.opengeospatial.org>. Access: October, 2012.
- ISO. *SQL Multimedia and Application Packages – Part 3: Spatial.*
- PostGIS: <http://postgis.net/>