

Instituto Superior Politécnico Gaya

Escola Superior de Ciência e Tecnologia

Informática de Gestão

Base de dados

2009/2010

*PostGIS - spatial database extension for
PostgreSQL*

**Joel Ricardo
César Pinto**

***Docente:* José Manuel da Costa Correia**

15 De Janeiro de 2010

Índice

Lista de Siglas	ii
Introdução	iii
Desenvolvimento	iv
• Definição de SIG	iv
• O PostgreSQL.....	6
-Acerca	6
• O PostGIS.....	7
-Acerca	7
-Tipos de dados suportados	7
Point.....	8
Line Segments.....	9
Boxes.....	10
Paths	10
Polygon	11
Circles.....	12
GeometryCollection.....	12
-Aplicações que usam PostGIS	13
• SQL e dados espaciais	14
Conclusões	17
Referencias Bibliográficas	18

Lista de Siglas

SIG – Sistema de informação geográfica

GIF – Geographic Information System

OGC – Open Geospatial Consortium

SQL – Structured Query Language

GEOS – Geometry Engine, Open Source

GPL – General Public License

Introdução

As bases de dados são um dos componentes mais utilizados na computação. Elas armazenam dados de diversos tipos que sozinhos não transmitem qualquer informação. Uma vez conjugados, estes dados tornam-se informação que pode corresponder a vários graus de exigibilidade de acordo com o utilizador.

As bases de dados vieram facilitar muito o trabalho das empresas e particulares que pretendem armazenar informação, e com ela criar dados altamente estruturados.

Este trabalho tratará o PostGIS, uma extensão do PostgreSQL que tem como função armazenar dados espaciais. É também abordado neste trabalho, as funcionalidades, as suas aplicações e as boas práticas de utilização desta extensão.

Com este trabalho pretende-se que um utilizador com conhecimentos de SQL consiga compreender e posteriormente aplicar a nova extensão PostGIS.

Desenvolvimento

- **Definição de SIG**

Um SIG ou extensivamente falando sistema de informação geográfica, é a junção de hardware, software, procedimentos computacionais e informação espacial. Uma vez formado o SIG, permite a representação, análise e gestão do espaço geográfico bem como todos os fenómenos que nele ocorrem.

Na parte do software, um SIG é apoiado por uma base de dados relacional que, bem estruturada, concisa e íntegra permite a eficiente manutenção dos dados que nela existam.

Um SIG separa os dados em camadas temáticas e armazena-os independentemente. Sendo assim, é possível trabalhar com os dados de modo rápido e eficiente com o intuito de gerar informação.

Cada SIG é implementado de acordo com um modelo. Os modelos mais comuns são o Modelo Raster ou Matricial¹ e o Modelo Vectorial. O modelo Raster tem como base a dimensão de um determinado espaço, compartimentando-o em células e organizando cada célula por ordem de precisão. Neste modelo quanto maior for a célula menos precisão vamos conseguir obter numa futura representação. Num SIG, a implementação do modelo Raster é feita com imagens em bitmap, porque são imagens constituídas através de um algoritmo tão complexo que conseguem guardar a informação única de cada pixel. Assim, são as imagens adequadas para este modelo porque conseguimos a máxima precisão de um pixel.

Enquanto no modelo matricial, a precisão é um elemento geral que se distribuir de acordo com o tamanho das suas células, no modelo vectorial, esta mesma precisão é atribuída de acordo com a localização dos elementos no espaço, ou seja, onde existirem mais elementos geográficos representados a precisão deve ser maior.

Sabendo já que cada modelo de SIG tem as suas características, é necessário saber que todos eles, para modelar digitalmente os objectos do mundo real e para representar no mapa as suas características, usam como padrão três elementos

¹ Modelo em que o PostGIS é implementado

espaciais para o efeito: o ponto, a linha e o polígono. Estes são tipos de dados essenciais para as representações que são precisas num SIG.

Com o desenvolvimento dos SIG's e o aparecimento dos citados tipos de dados, foi necessário criar padronizações, métodos genéricos de dados que permitissem a uniformização da informação. São elas:

- WMS – Web Map Service
- WFS – Web Feature Service
- WCS – Web Coverage Service
- CS-W – Catalog Service Web
- SFS² – Simple Features Interface Standard
- GML – Geography Markup Language

Todas as normas ou especificações anteriormente referidas são promovidas pela OGC que voluntariamente tenta regular e criar padrões para todo o conteúdo e serviços geográficos. A OGC é constituída por mais de 280 organizações de todos os tipos e de todo o mundo que trabalham regularmente para a já falada uniformização.

O PostGIS, o foco fundamental deste trabalho, trabalha segundo a norma SFS. Esta oferece um programa bem definido e uma forma de trabalho comum para aplicações poderem aceder a bases de dados relacionais. Tem também como função o suporte de acesso a dados geográficos guardados na base de dados. Esta norma ou especificação divide-se basicamente em duas partes:

- Uma primeira parte com uma arquitectura genérica, que fornece o modelo característico comum de interface e acesso aos dados.

- Uma segunda parte que fornece a implementação padrão SQL³ para a comunicação com a primeira parte.

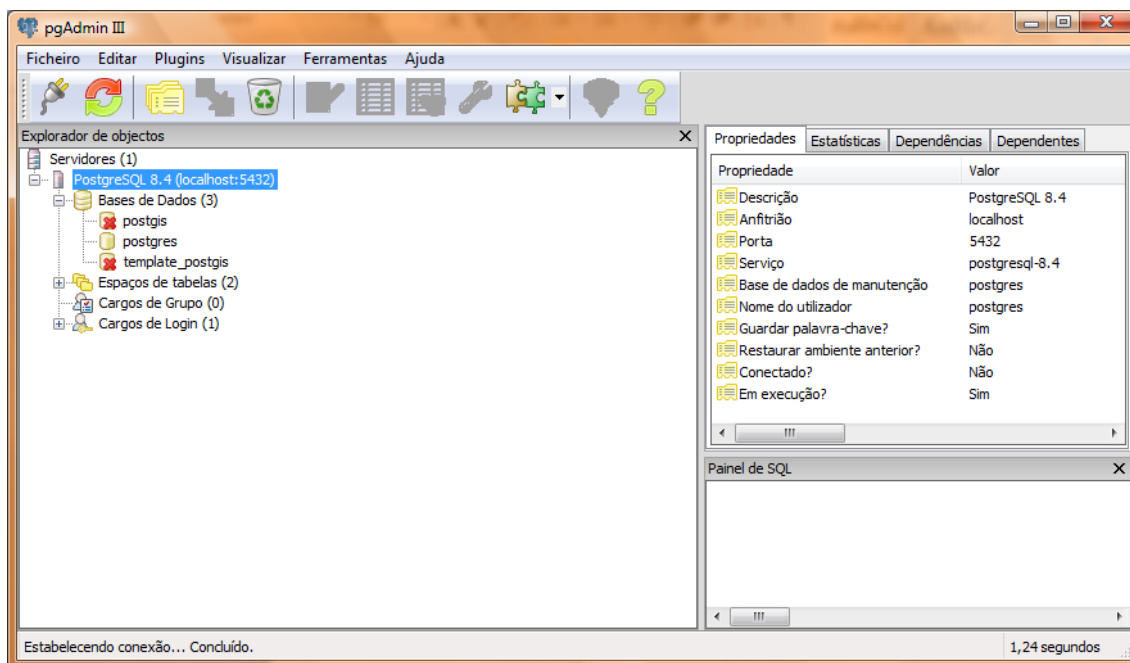
² Norma que o PostGIS usa disponibilizado pela OGC.

³ Linguagem inspirada na álgebra relacional, tem como objectivo, a consulta estruturada de dados de uma base de dados relacional.

- **O PostgreSQL**

-Acerca

O PostgreSQL é um sistema de gestão base de dados relacional. Conhece-se como uma ferramenta grátis que funciona em todos os principais sistemas operativos. Com mais de 15 anos de existência, este SGBD conseguiu uma enorme confiabilidade entre o mundo dos programadores. Consegue-se integrar com o c#, c++, c, perl, etc...



1. SGBD PostgreSQL

- **O PostGIS**

-Acerca

O PostGIS, em constante desenvolvimento pela Refractions Research Inc, é uma poderosa extensão do acima referido PostgreSQL. Ela adiciona um suporte para objectos espaciais às bases de dados relacionais permitindo que estas consigam guardar dados do tipo geográfico/espacial numa base de dados relacional.

Esta extensão foi desenvolvida ao abrigo da licença GPL (General Public License) e é considerada completamente grátis.

Neste momento as linguagens mais importantes como c#, c, perl, php, java, etc já conseguem manobrar os dados espaciais de forma eficiente e relativamente rápida. A representação dos dados geográficos é feita através de API's que ajudam os softwares a interpretar os dados armazenados.

O PostGIS usa a biblioteca de software GEOS⁴ para efectuar todos os cálculos geoespaciais necessários, tais como, áreas, comprimentos, distancias, larguras ou perímetros. Esta livraria trabalha em simultâneo com a norma SFS para disponibilizar as funções e operadores espaciais necessários para o trabalhar dos dados geográficos.

A implementação do PostGIS é considerada “leve” porque devido aos seus índices e geometrias optimizadas, consegue reduzir o uso de disco e memória do sistema onde se encontra implementado. Com esta qualidade, o PostGIS consegue usar de forma eficiente os recursos do sistema e assim construir dados de qualidade melhorando os resultados de pesquisa.

-Tipos de dados suportados

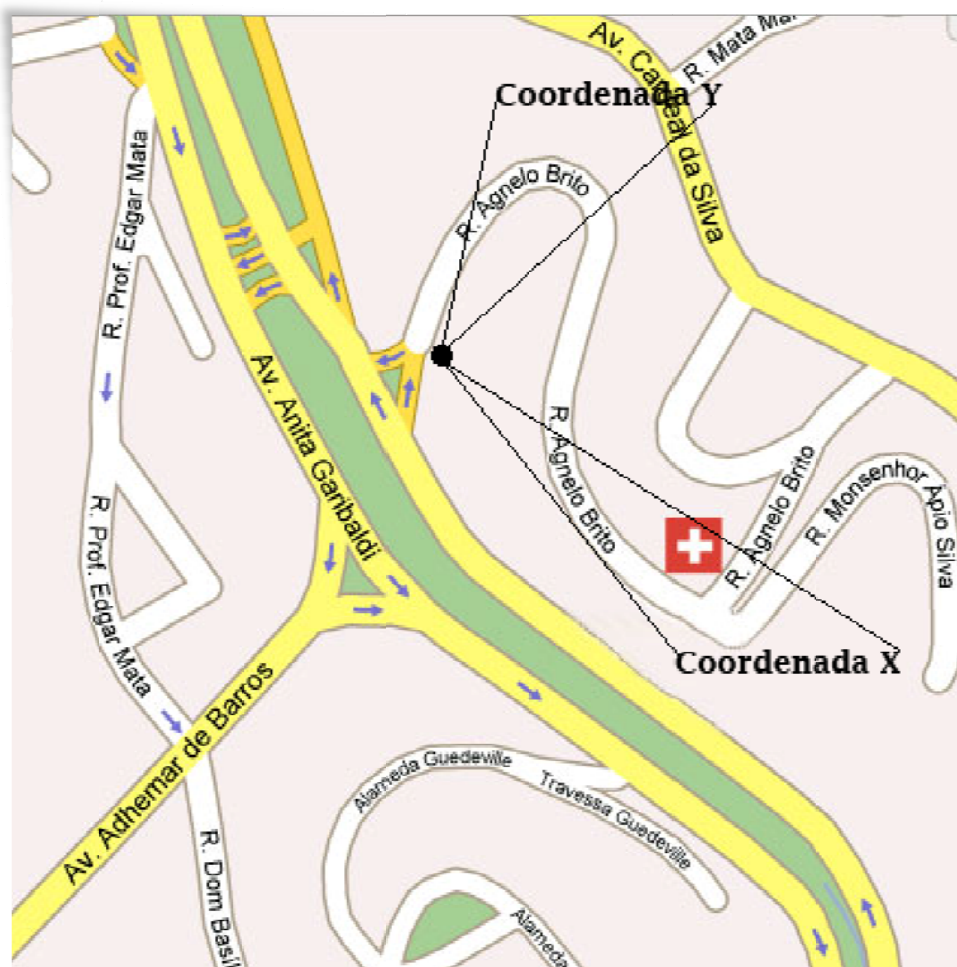
Existem tipos de dados específicos do PostGIS, que têm como função armazenar os dados espaciais de forma arrumada e o mais simples possível para que estes ocupem o menor espaço possível na base de dados e facilitem breves pesquisas.

⁴ Livraria de software que contém todas as funções geométricas de integração com o SQL

Ponto (Point)

Este tipo de dados armazena dois dados/coordenadas que como o próprio nome indica localizam um ponto único.

Sintaxe: (x , y)

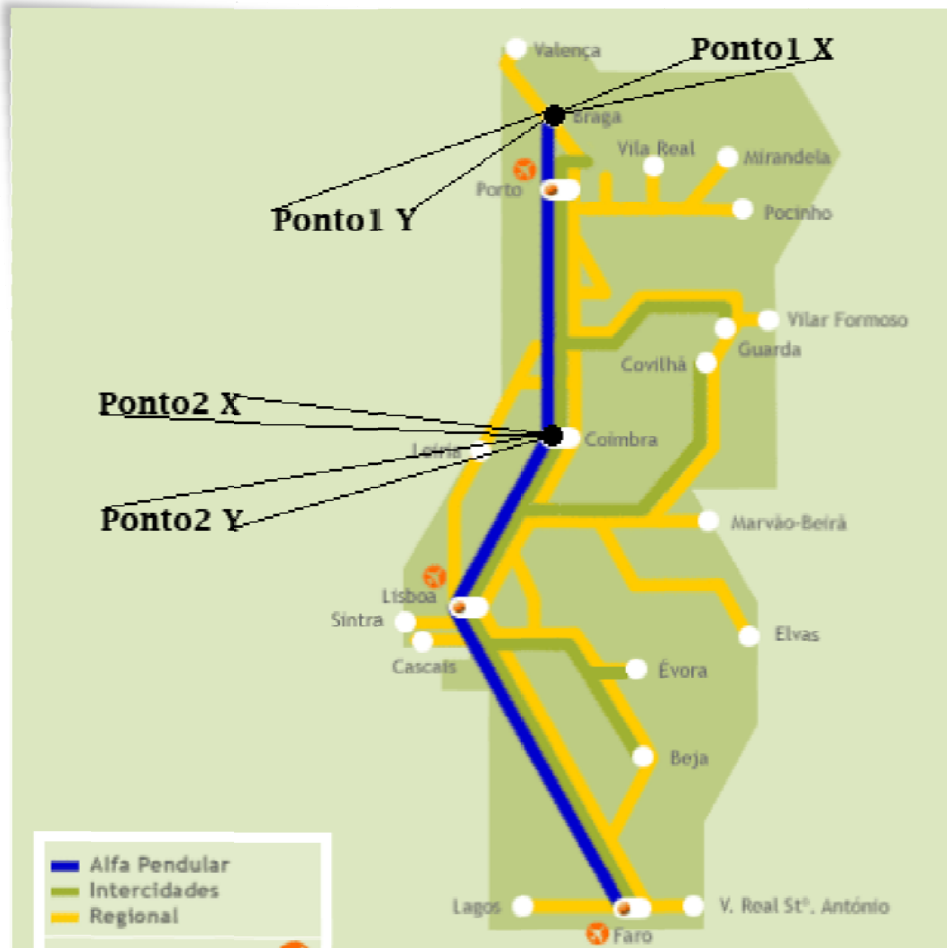


Fonte: <http://ssbb.ch/wp-content/uploads/2007/10/mapa.jpg>

Linha (Line Segments)

Representados por pares de pontos, este tipo de dados é utilizado para traçar linhas georreferenciadas.

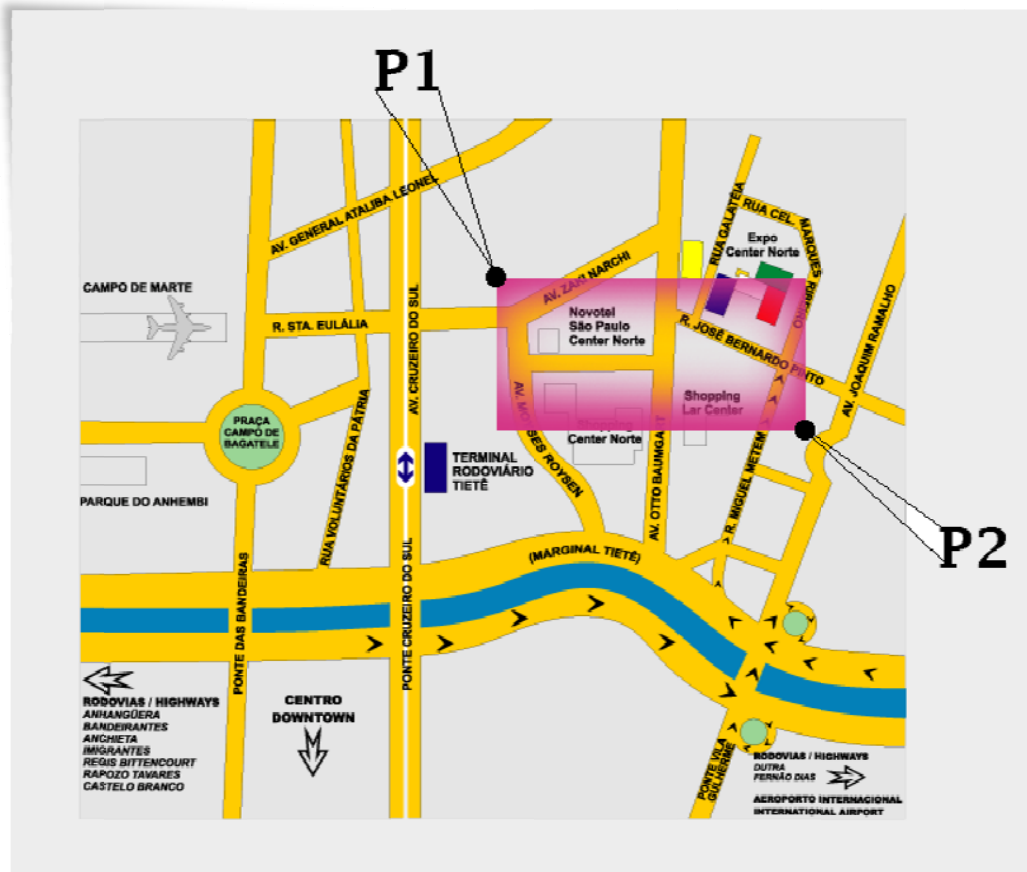
Sintaxe: ((x1 , y1) , (x2 , y2))



Fonte: http://www.cp.pt/StaticFiles/Imagens/Mapas/Viajar%20em%20Portugal/mapa_pt.gif

Caixas (Boxes)

Também representado por pares de pontos, este tipo de dados armazena dados equivalentes aos cantos opostos de uma caixa. Segue um esquema para melhor compreensão:



Fonte: <http://www.asuf.net/mapa.gif>

Assim é armazenado o P1 e o P2. Com uma subrotina já predefinida no PostGIS, é calculada a caixa de referência geográfica.

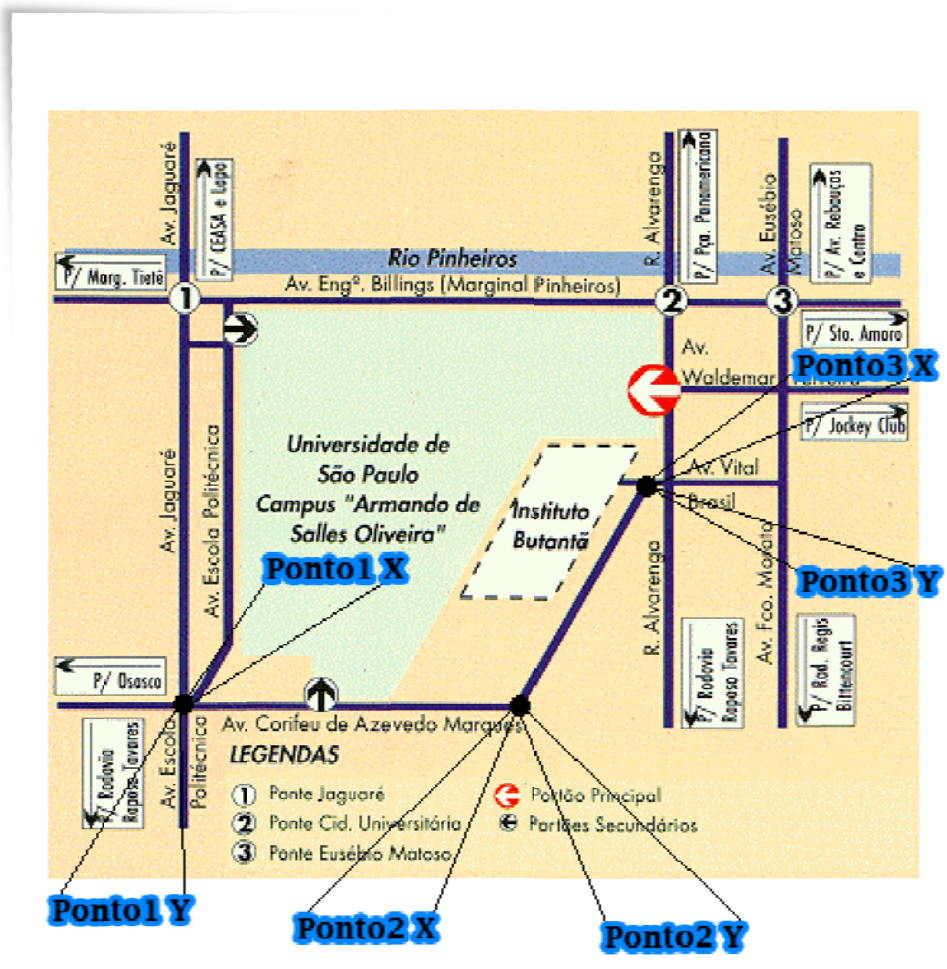
Syntaxe: $((x_1, y_1), (x_2, y_2))$

Caminhos (Paths)

Em português designados por caminhos, este tipo de dados armazena pontos, ou seja, (x,y) . O conjunto destes pontos forma um caminho que poderá ser interpretado e representado posteriormente. Deve-mos ter em atenção que

este tipo de dados nunca fecha o caminho, ou seja, mesmo que os pontos se sobreponham, $((1,1)(1,1) \dots (1,1)(1,1))$, nunca é considerado como um polígono, para isso existe um tipo de dados chamado Polygon que será abordado mais á frente.

Sintaxe: $((x_1, y_1), \dots, (x_n, y_n))$



Fonte: <http://www.iea.usp.br/iea/imagens/mapa1.gif>

Polígonos (Polygon)

Este tipo de dados armazena listas de pontos, correspondentes a vértices, que uma vez juntos criam uma forma geométrica fechada, designada polígono. Tem muitas parecenças com o caminho paths, mas as suas rotinas de suporte e o seu modo lógico de armazenamento são completamente diferentes.



-Aplicações que usam PostGIS

Existem muitas aplicações de serviços geográficos que usam o PostGIS. Algumas delas são:

- Everest GIS⁵
- GRASS GIS (GPL)⁶
- gvSIG (GPL)⁷
- TerraView (GPL)⁸
- MapGuide (LGPL)⁹
- MapServer (BSD)¹⁰

Todas estas aplicações e muitas mais tentam usar todas as potencialidades que o PostGIS oferece gratuitamente abrangido pela norma GPL.

⁵ Software pago de mapeamento (<http://www.everest-gis.com/>)

⁶ Ferramenta gratuita mais utilizada mundialmente para SIG's (<http://grass.itc.it/>)

⁷ Aplicação de origens espanholas, também esta de mapeamento geográfico (<http://www.gvsig.gva.es/>)

⁸ Software SIG que funciona com diversas bases de dados, como por exemplo Access e PostGIS (<http://www.dpi.inpe.br/terraview/index.php>)

⁹ Aplicação com a possibilidade de integração em Windows ou Linux (<http://mapguide.osgeo.org/>)

¹⁰ Plataforma de publicação de dados geográficos (<http://mapserver.org/>)

- **SQL e dados espaciais**

O PostgreSQL, com a extensão PostGIS, passou ele também a poder manobrar dados geográficos. Devido á forma como o SQL está estruturado, é possível o armazenamento de dados geográficos e posteriormente o seu tratamento. Estes dados uma vez relacionados transformam-se em informação e amostragens geográficas.

- **Integração de SQL com PostGIS**

Para demonstrar o SQL e os comandos que interagem com o PostGIS, será apresentado um pequeno exemplo constituído por somente uma tabela. Esta tabela terá informação relativa a pontos de interesse gerais.

Passo 1: Criação da base de dados

SQL:

```
CREATE DATABASE pontos_interesse
```

Passo 2: Criação da tabela de referências

CAMPOS:

- | | |
|--------------|---|
| 1. srid | //Referência única de identificação SRS ¹¹ |
| 2. auth_name | //Nome do standard utilizado |
| 3. auth_srid | //Valor do SRS |
| 4. srtext | //Representação em texto da SRS |
| 5. proj4text | //Identificador da livreria a utilizar |

¹¹ Spatial Referencing System

SQL:

```
CREATE TABLE spatial_ref_sys (  
  srid    INTEGER NOT NULL PRIMARY KEY,  
  auth_name VARCHAR(256),  
  auth_srid INTEGER,  
  srtext  VARCHAR(2048),  
  proj4text VARCHAR(2048)  
)
```

Passo 3: Criação da tabela geométrica

CAMPOS:

- f_table_catalog //Nome do catalogo onde está a informação da base de dados. Encontra-se no Admin do PostGRE
- f_table_schema //Nome do esquema da base de dados. Por defeito denomina-se “public”
- f_table_nam // Nome da tabela geométrica
- f_geometry_column //Nome da coluna geométrica
- coord_dimension //Dimensão especial da coluna (2, 3 ou 4)
- srid //Id da coordenada geométrica referente ao registo nesta tabela. Campo chave estrangeira.
- type //Tipo do objecto espacial. Por ex: “POINT”

SQL:

```
CREATE TABLE geometry_columns (  
  f_table_catalog VARCHAR(256) NOT NULL,  
  f_table_schema VARCHAR(256) NOT NULL,  
  f_table_nam VARCHAR(256) NOT NULL,  
  f_geometry_column VARCHAR(256) NOT NULL,  
  coord_dimension INTEGER NOT NULL,  
  srid    INTEGER NOT NULL,  
  type    VARCHAR(30) NOT NULL  
)
```


Passo 4: Criação da tabela espacial

Passo 4.1: Criar uma tabela normal

CAMPOS:

- Id //Identificador único do registo
- nome //Nome do ponto geográfico
- descricao //Descrição do ponto geográfico

SQL:

```
CREATE TABLE pontos_interesse (  
id int4 NOT NULL PRIMARY KEY,  
nome varchar(25),  
descricao varchar(1000)  
)
```

Passo 4.2: Adicionar a coluna geográfica

ESQUEMA:

```
AddGeometryColumn(  
  <schema_name>,  
  <table_name>,  
  <column_name>,  
  <srid>,  
  <type>,  
  <dimension>  
)
```

SQL:

```
SELECT AddGeometryColumn('public', 'pontos_interesse', 'geom', 3763,  
'POINT',0)
```

Passo 5: Inserir registo

SQL:

```
INSERT INTO pontos_interesse (geom, nome, descricao )  
VALUES (ST_GeomFromText('POINT(-7.845 39.58)', 2963), 'Lisboa', "Capital  
Portuguesa");
```

Conclusões

• Aprendizagem adquirida

Durante o desenvolvimento deste trabalho foram pesquisadas matérias que interessavam e outras que não interessavam para o assunto apresentado. No entanto, tudo foi informação nova e importante para o aumento da sabedoria particular. Este trabalho apresenta uma ferramenta poderosa de bases de dados, com a vantagem de ser completamente grátis. Foi possível aprender razoavelmente bem a ferramenta. Comunicar com a ferramenta e interagir sobre ela foram os principais feitos ao desenvolver este trabalho. Foi necessário aprofundar o SQL, visto que esta é a linguagem com que a ferramenta trabalha.

Falando agora de dados geométricos, muitos destes dados são comuns também aos outros tipos de bases de dados concorrentes. Assim, sabendo os tipos que o PostGIS suporta, sabe-se o básico para interagir com as outras ferramentas concorrentes.

• Dificuldades ultrapassadas

Uma das principais dificuldades foi estruturar a informação de modo a ser facilmente compreendida pelo público-alvo. Existe muita informação sobre PostGRE e PostGIS, mas não tão coerente como se precisa.

Ultrapassado o problema na coerência de informação, deparamo-nos com a passagem da teoria para a prática. Agindo no ambiente de Admin do PostGRE, foi tratado o SQL de forma a construir uma pequena base de dados composta por somente três tabelas. A dificuldade estava na inserção da coluna geométrica na tabela “pontos_interesse” que, não funcionava pelo simples facto do servidor estar desligado.

Vendo estes dois problemas ultrapassados, concluiu-se o estudo do PostGIS, a base de dados capaz de armazenar e tratar dados geográficos.

Referencias Bibliográficas

1. <http://www.postgresql.org/docs/8.1/static/datatype-geometric.html#AEN5194>
2. http://pt.wikipedia.org/wiki/Sistema_de_informação_geográfica
3. http://pt.wikipedia.org/wiki/Sistema_de_informação_geográfica#Modelos
4. <http://pt.wikipedia.org/wiki/SQL>
5. <http://spatialreference.org>
6. <http://theworldofapenguin.blogspot.com/2008/06/circles-in-postgis.html>

Anexos

Nesta área encontra-se somente material de apoio ao trabalho principal sobre os temas PostGRE e PostGIS.

Este trabalho é exclusivamente para uso académico e focalizado para a disciplina de Base de Dados.

