

Mapserver e Bancos de Dados

João Araujo



O que é um Banco de Dados?

- É uma ferramenta usada para armazenar e acessar tabelas de informação.
- O tipo de dados pode variar de acordo com o Banco de Dados, mas tipicamente armazena números e texto.

Banco de Dados Relacional

- Um Banco de Dados Relacional ou Base de Dados Relacional, ou ainda um Sistema Gerenciador de Banco de Dados Relacional (SGDBR) é um sistema de armazenamento de dados baseado no princípio em que todos os dados estão armazenados em tabelas (relações), seguindo o **Modelo Relacional**.
-
-

Banco de Dados Espaciais

- É um banco de dados que armazena informação geográfica.
- Vários produtos do mercado usam o termo com diferenças sutis no seu significado.

Produtos

- ESRI's Spatial Database Engine (**SDE**) – Que não é, estritamente falando, um BDE, mas uma interface entre o cliente e um banco de dados convencional, onde são armazenados dados espaciais no formato ESRI.
 - **Oracle** Spatial
 - Extensão Espacial para o banco de dados **DB2** da IBM.
 - Extensão Espacial para **MySQL**.
 - **PostGIS**
-
-

PostGIS

- Uma extensão do PostgreSQL.
 - Banco de Dados Espaciais Avançado.
 - Pode armazenar e manipular Dados Espaciais.
 - Não é simplesmente um depósito de dados, mas um ambiente para interagir com dados espaciais.
 - Implementação robusta de todas as especificações da OGC Simple Features Specification for SQL (SFSQL).
 - O tipo e as coordenadas do objeto são definidos segundo o OGC WKT (Well-Known Text) e WKB (Well-KnownBinary).
-
-

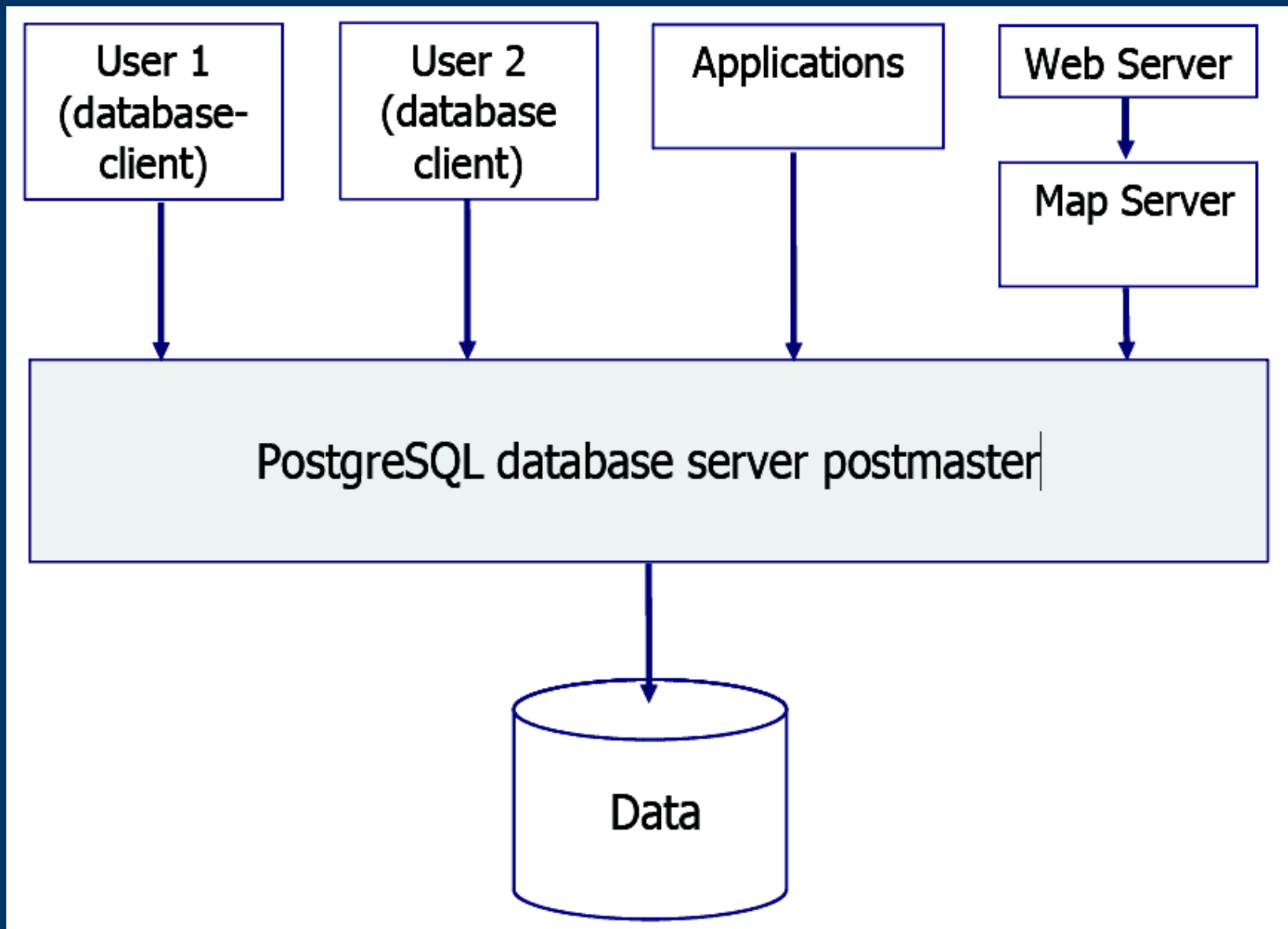
Exemplos WKT como definidos pelo OGC

- **POINT**(2572292.2 5631150.7)
 - **LINESTRING** (2566006.4 5633207.9, 2566028.6 5633215.1, 2566062.3 5633227.1)
 - **MULTILINESTRING**((2566006.4 5633207.9, 2566028.6 5633215.1), (2566062.3 5633227.1, 2566083 5633234.8))
 - **POLYGON** (2568262.1 5635344.1, 2568298.5 5635387.6, 2568261.04 5635276.15, 2568262.1 5635344.1);
-
-

PostGreSQL

- Também chamado de Postgres.
- Brasileiros também chamam de Postgree(?).
- Banco de Dados Objeto-Relacional.

PostgreSQL é organizado segundo o modelo cliente-servidor



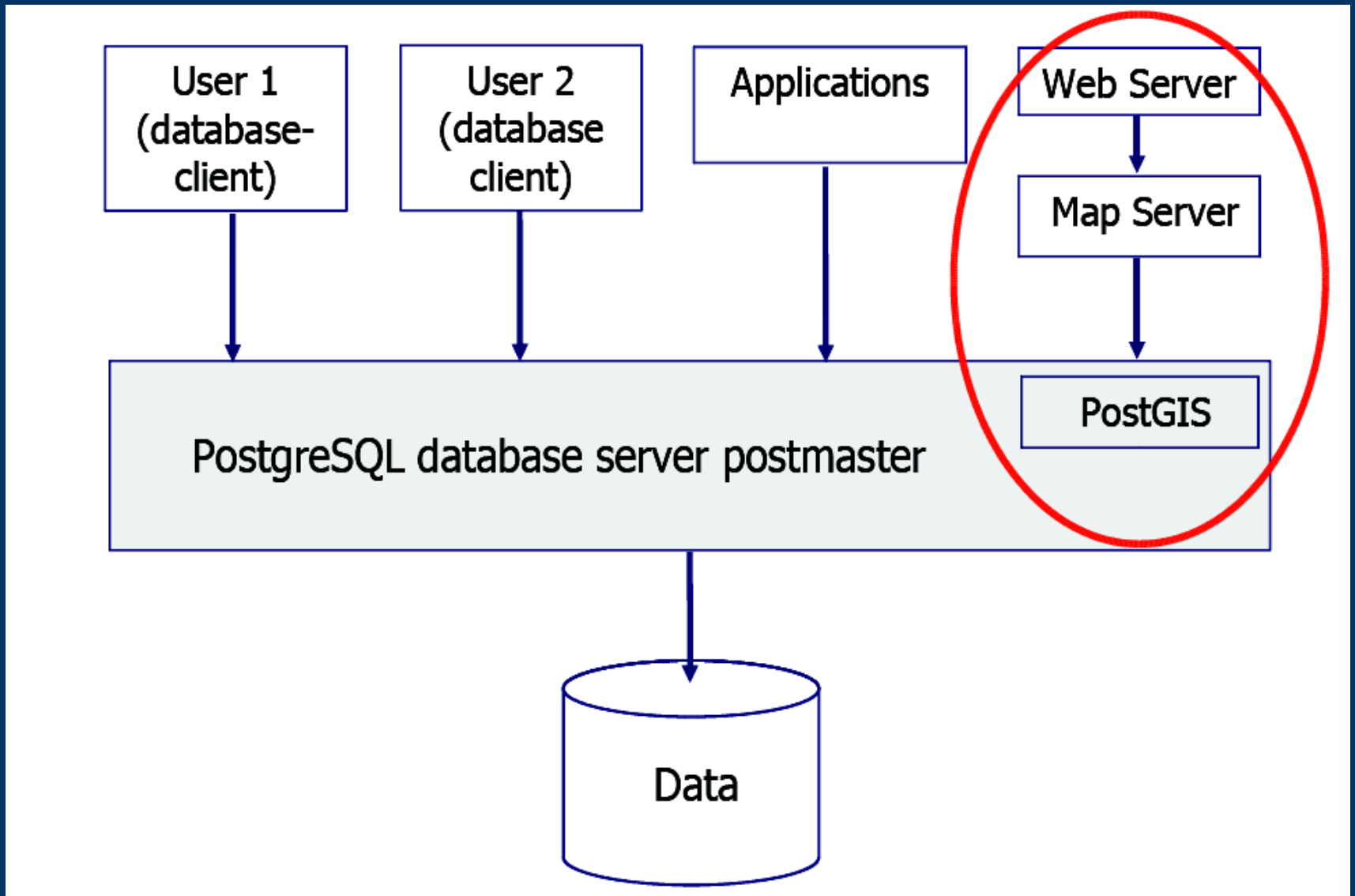
Modelo Cliente-Servidor do PostgreSQL - Servidor

- Um **processo servidor**, que gerencia os arquivos de banco de dados, recebe conexões dos aplicativos cliente com o banco de dados, e executa ações no banco de dados em nome dos clientes.
 - O programa servidor de banco de dados se chama **postmaster**.
-
-

Modelo Cliente-Servidor do PostgreSQL - Cliente

- O aplicativo **cliente** do usuário (frontend) que deseja executar operações de banco de dados.
 - O cliente pode ser uma ferramenta no modo caractere, um aplicativo gráfico, um servidor Web que acessa o banco de dados para mostrar páginas Web, ou uma ferramenta especializada para manutenção do banco de dados.
-
-

PostGIS é um módulo de extensão do PostgreSQL



Consulta ao BD

- O servidor processa o pedido, monta os dados e envia o resultado.
 - Todo o trabalho é feito no servidor, pois a maioria dos computadores não está preparada para a carga da consulta aos BD.
 - Se todo o dado tiver que ser enviado pela rede para o cliente, a rede e o cliente passam a ser o gargalo do sistema.
-
-

Aplicações GIS

- Usam o poder de computação da máquina cliente.
 - Exemplo: a fusão de vários polígonos baseada em algum atributo.
 - A aplicação carrega todos os polígonos na memória ou em algum arquivo temporário.
 - Quando o processo começa, o uso do processador e da memória pode ser intenso.
 - Se a aplicação não possui a funcionalidade requerida, você tem que comprar algum plugin ou aplicação alternativa.
-
-

Futuro das Aplicações GIS

- Serão apenas pouco mais que visualizadores, com toda a funcionalidade acontecendo no banco de dados espaciais.



PostGIS

- Implementa o OGC Simple Feature Specifications for SQL (SFSQL).
 - OGC define operações para inserir, consultar, manipular e deletar objetos espaciais.
 - As coordenadas do objeto são armazenadas em tabelas de feições.
-
-

PostGIS

Possui três componentes:

- Biblioteca libpostgis que é a interface entre PostgreSQL e as capacidades espaciais do PostGIS.
 - Um script principal para carregar funções e tipos do PostGIS.
 - Um script opcional para trabalhar com projeções.
-
-

PostGIS

Site: <http://postgis.refractory.net/docs/>

Interfaces do PostGIS para Dados GIS.

- Nativamente possui suporte para shapefile.
 - Vários tipos de dados (Shape, MapInfo, DGN, GML, ...) podem ser lidos, convertidos e inseridos no PostGIS com auxílio das bibliotecas OGC.
-
-

GEOS

- GEOS (Geometry Engine - Open Source) provê um modelo de objetos espaciais além de funções geométricas fundamentais.
 - Funções fornecidas incluem: predicados espaciais, funções de intersecção, diferença, união etc.
 - Escrita em C++.
-
-

PROJ

- Biblioteca usada para reprojetar ou transformar coordenadas.



Conexão

- O **servidor PostgreSQL** pode tratar várias conexões simultâneas de clientes.
 - É iniciado um novo processo (**fork**) para cada conexão.
 - Daí em diante, o cliente e o novo processo servidor se comunicam sem intervenção do processo **postmaster** original.
 - O **postmaster** está sempre executando aguardando por novas conexões dos clientes.
-
-

Instalando PostgreSQL, PostGIS e Geos no Ubuntu.

- Pode-se usar o synaptic ou
- Instalando na mão, usando os pacotes mais recentes.



Passos

- Instalar executar PostgreSQL
 - Habilitar a linguagem de suporte pl/pgsql
 - Carregar o script postgis.sql (ou lwpostgis.sql)
 - Carregar o script spatial_ref_sys.sql
 - Criar o banco de dados
-
-

1: Baixar Arquivos

- Baixar o PostgreSQL:
 - <http://www.postgresql.org/download/>
 - Baixar o Geos:
 - <http://geos.refrations.net/>
 - Baixar o PostGIS:
 - <http://postgis.refrations.net/download/>
-
-

2: Instalar pacotes

```
sudo apt-get install build-essential  
libgeos-dev python2.4-dev libreadline5-  
dev libperl-dev flex byacc proj zlib1g-  
dev
```



3: Extrair Arquivos

```
tar -xf geos-2.2.1.tar.bz2
```

```
tar -xf postgresql-8.1.1.tar.bz2
```

```
tar -xf postgis-1.1.0.tar.gz
```



4: Mover PostGIS para contrib

```
mv postgis-1.1.0 postgresql-8.1.1/contrib/
```



5: Instalar Geos

```
cd geos-2.2.1  
./configure  
make  
sudo make install
```



6: Instalar postgresql para /gis

```
cd ../postgresql-8.1.1
./configure --prefix=/gis --with-perl --with-
python
make
sudo make install
```

7: *Configurar PostGIS*

```
cd contrib/postgis-1.1.0/  
./configure --with-pgsql=/gis/bin/pg_config  
  
# remover ligações experimentais do  
  geos  
echo USE_GEOS_CAPI=0 >>  
  Makefile.config  
make  
sudo make install
```

Criar o BD

```
# criar um usuário para o postgresql  
sudo adduser gisuser
```

```
# criar o banco de dados  
sudo mkdir /gis/data  
sudo chown gisuser /gis/data  
su - gisuser  
/gis/bin/initdb -D /gis/data
```

Iniciar o PostgreSQL

```
/gis/bin/pg_ctl -D /gis/data -l  
postgresql.log start
```



Testando a Instalação do PostgreSQL

```
> psql -l
```

Lista dos bancos de dados

```
Nome | Dono | Codificação
```

```
-----+-----+-----
```

```
template0 | postgres | LATIN1
```

```
template1 | postgres | LATIN1
```

```
(2 registros)
```

Para testar em outro computador:

```
> psql -l -h servidor.com
```

Para usar outro usuário:

```
> psql -l -h myserver.com -U postgres
```

Habilitando pl/pgsql

- PostgreSQL tem suporte para uma linguagem de programação interna que pode interagir e ser incorporada ao banco de dados.
 - Esta linguagem não é habilitada por default.
 - Deve ser habilitada explicitamente:
`createlang plpgsql template1`
-
-

Carregando o PostGIS no PostgreSQL

- Para habilitar o PostGIS devemos carregar as suas funções no PostgreSQL.
 - Se as funções forem carregadas em `template1`, todos os bancos de dados criados a partir daí terão as funcionalidades do PostGIS.
- ```
> psql -f lwpostgis.sql -d testgis
```
- 
-

# *Carregando o script spatial\_ref\_sys.sql*

- A tabela `spatial_ref_sys.sql` possui informações sobre projeções.
  - Com ela, o PostGIS é capaz de fazer reprojeções.
- ```
> psql -f spatial_ref_sys.sql -d testgis
```



Testando a funcionalidade do PostGIS

> `psql testgis`

Bem-vindo ao psql 8.0.3, o terminal iterativo do PostgreSQL.

Digite: `\copyright` para mostrar termos de distribuição

`\h` para ajuda com comandos SQL

`\?` para ajuda com comandos do psql

`\g` ou terminar com ponto-e-vírgula para executar a consulta

`\q` para sair

testgis=#

Testando se as tabelas foram criadas

```
testgis=# \dt
```

Lista de relações

Esquema	Nome	Tipo	Dono
public	geometry_columns	tabela	postgres
public	spatial_ref_sys	tabela	postgres

(2 registros)

Consulta

```
testgis=# SELECT count(*) FROM  
    spatial_ref_sys;
```

```
count
```

```
-----
```

```
2671
```

```
(1 registro)
```



Conferindo algumas funções do PostGIS

```
testgis=# \df postgis*
```

Lista de funções			
Esquema	Nome	Tipo de dado do resultado	Tipos de dado do argumento
public	postgis_full_version	text	
public	postgis_geos_version	text	
public	postgis_gist_joinsel	double precision	internal, oid, internal, smallint
public	postgis_gist_sel	double precision	internal, oid, internal, integer
public	postgis_jts_version	text	
public	postgis_lib_build_date	text	
public	postgis_lib_version	text	
public	postgis_proj_version	text	
public	postgis_scripts_build_date	text	
public	postgis_scripts_installed	text	
public	postgis_scripts_released	text	
public	postgis_uses_stats	boolean	
public	postgis_version	text	

(13 registros)

Conferindo versão do PostGIS

```
testgis=# SELECT postgis_version( );  
          postgis_version
```

1.1 USE_GEOS=1 USE_PROJ=1
 USE_STATS=1
(1 registro)

Criando Banco de Dados

```
> createdb project1
```

```
CREATE DATABASE
```

```
> psql -l
```

Lista dos bancos de dados

Nome	Dono	Codificação
------	------	-------------

-----+-----+-----

project1	postgres	LATIN1
----------	----------	--------

template0	postgres	LATIN1
-----------	----------	--------

template1	postgres	LATIN1
-----------	----------	--------

testgis	postgres	LATIN1
---------	----------	--------

(4 registros)

Preenchendo o BDE

- Podemos usar comandos SQL com psql ou
- Usar ferramentas que exportam dados diretamente no BD.



Usando shp2pgsql

- PostGIS também tem seu inverso: pgsql2shp.
 - Convertem dados de shapefile para postGIS e vice-versa.
 - shp2pgsql transforma um shapefile numa seqüência de comandos SQL. Com isto, é necessário direcionar sua saída para um arquivo texto para depois incorporá-lo ao BDE.
 - `shp2pgsql countyp020.shp countyp020 > mycounties.sql`
 - `psql -d project1 -f mycounties.sql`
-
-

Usando ogr2ogr

- `ogr2ogr -f "PostgreSQL"
PG:dbname=project1 countyp020.shp`



FIM

