

BASES DE DATOS ESPACIALES

Introducción a PostGIS

PostgreSQL – PostGIS

¿Qué es PostGIS?

Es una extensión al sistema manejador de bases de datos relacional PostgreSQL. Permite el uso de objetos GIS (Geographic Information Systems).

PostGIS incluye soporte para índice GiST basados en R-Tree y funciones básicas basadas en objetos GIS.

Con PostGIS se pueden usar todos los objetos que aparecen en la especificación OpenGL como:

Líneas

Multilíneas

Colecciones geométricas

Puntos

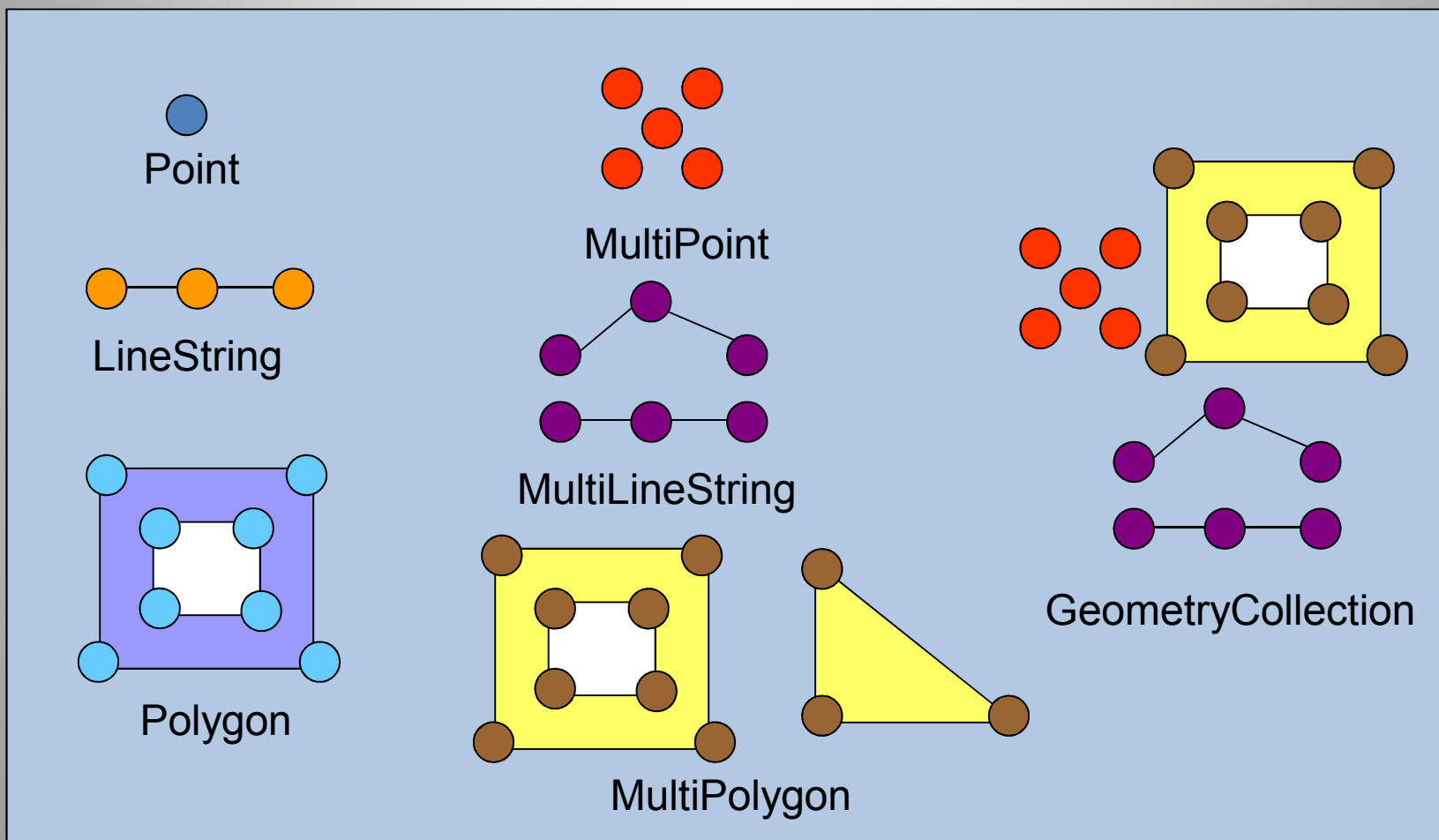
Multipuntos

Polígonos

Multipolígonos

Base de Datos Espaciales

Tipos de Datos Espaciales



Base de Datos Espaciales


OpenGIS WKB y WKT

Ejemplos de Well – Know Text (WKT)



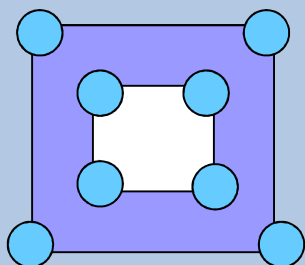
Point

POINT (0 0)



LineString

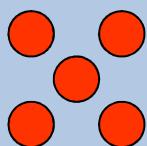
LINESTRING (0 0,1 0, 2 0)



POLYGON((0 0,0 3,3 3,3,0),(1 1,1 2,2 2,2 1))

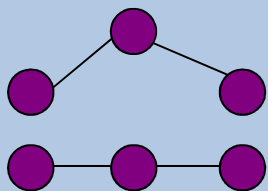
Base de Datos Espaciales

OpenGIS WKB y WKT



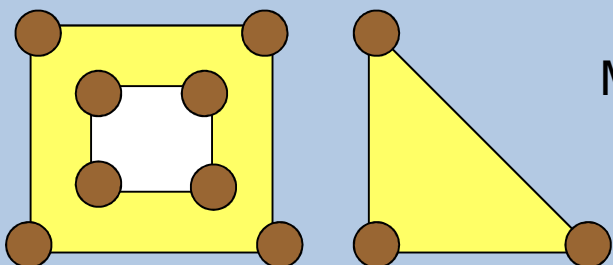
MultiPoint

MULTIPOINT (0 0,0 2,1 1,2 2,2 1)



MultiLineString

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

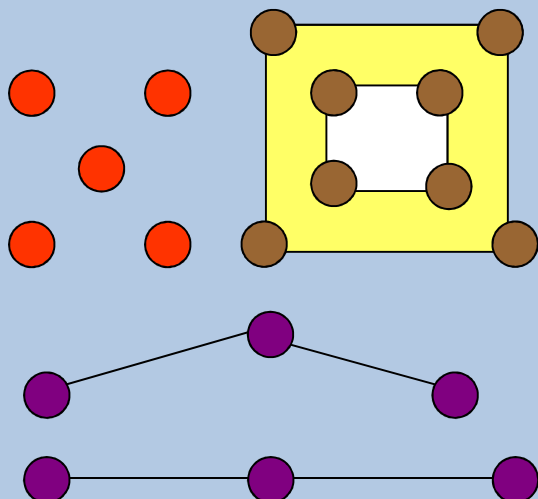


MultiPolygon

MULTIPOLYGON(((0 0,0 3,3 3,3,0),(1 1,1 2,2 2,2 1)),
((4 0,4 3,7 0)))

Base de Datos Espaciales

OpenGIS WKB y WKT



GeometryCollection

```
GEOMETRYCOLLECTION (  
  MULTIPOINT (0 3,0 5,1 4,2 5,2 4)),  
  MULTILINESTRING ((0 0,1 0, 2 0),  
                    (0 2,1 3,2 2))  
  MULTIPOLYGON((3 3,3 6,6 6,6,3),  
                (4 4,4 5,5 5,5 4)))
```

Base de Datos Espaciales

Usando estándares de Open Geospatial Consortium (OGC)

La especificación para SQL de características simples de la OGC define tipos de objetos GIS estándar, los cuales son manipulados por funciones, y un conjunto de tablas de metadatos.

Hay 2 tablas de meta-datos en la especificación OpenGIS:

- SPATIAL_REF_SYS
- GEOMETRY_COLUMNS

Base de Datos Espaciales

Usando estándares de Open Geospatial Consortium (OGC)

SPATIAL_REF_SYS

Contiene un identificador numérico y una descripción textual del sistema de coordenadas usados en la base de datos espacial.

Cumple con una lista aproximada de 3000 sistemas de referencia espacial, que pueden ser manejados, y también permite crear otros propios.

La definición de ésta tabla es la siguiente:

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


Base de Datos Espaciales

Usando estándares de Open Geospatial Consortium (OGC)

SPATIAL_REF_SYS

Las columnas de las tablas son la siguientes:

SRID: Valor entero que identifica el sistema de referencia espacial.

AUTH_NAME: El nombre del estándar para el sistema de referencia. Por ejemplo: EPSG.

AUTH_SRID: El identificador según el estándar AUTH_NAME. En el ejemplo anterior es el id según EPSG.

SRTEXT: Una *Well-know text* representación para el sistema de referencia espacial. Ejemplo: WKT para SRS.

PROJ4TEXT: Proj4 es una librería que usa PostGIS para transformar coordenadas . Esta columna contiene una cadena con definición de las coordenadas de Proj4 para un SRID dado.

Base de Datos Espaciales

Usando estándares de Open Geospatial Consortium (OGC)

GEOMETRY_COLUMNS

La definición de esta tabla es:

```
CREATE TABLE geometry_columns (  
    f_table_catalog VARRCHAR(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  
)
```

Base de Datos Espaciales

Usando estándares de Open Geospatial Consortium (OGC)

GEOMETRY_COLUMNS

Descripción de cada una de las columnas:

F_TABLE_CATALOG, F_TABLE_SCHEMA, F_TABLE_NAME: Distingue totalmente la tabla de características que contiene la columna geométrica.

F_GEOMETRY_COLUMN: Nombre de la columna geométrica en la tabla de características.

COORD_DIMENSION: Dimensión espacial de la columna(2D o 3D).

SRID: Es una clave foránea que referencia SPATIAL_REF_SYS.

TYPE: Tipo de objeto espacial. POINT, LINESTRING, POLYGON, MULTIPOINT, GEOMETRYCOLLECTION. Para un tipo heterogéneo se debe usar el tipo GEOMETRY.

Base de Datos Espaciales

Creando una tabla espacial

La creación de una tabla espacial se realiza en dos fases:

- Se crea una tabla no espacial
- Se añade una columna (campo) espacial a la tabla usando la función *AddGeometryColumn* de OpenGIS.

Sintaxis:

```
AddGeometryColumn(  
<NombreSchema>,  
<NombreTabla>,  
<NombreColumna>,  
<srid>,  
<Tipo>,  
<Dimensión>)
```

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

Base de Datos Espaciales

Creando una tabla espacial

La creación de una tabla espacial se realiza en dos fases:

Ejemplo

- CREATE TABLE geoCamino (id int4, nom varchar(25))
- SELECT AddGeometryColumn('public','geoCamino','geom', 423, 'LINESTRING', 2)

SELECT AddGeometryColumn('geoCamino', 'geom', 423, 'LINESTRING', 2)

La creación de una tabla espacial se realiza en dos fases:

Ejemplo

- CREATE TABLE geoCamino (id int4, nom varchar(25))
- SELECT AddGeometryColumn('public','geoCamino','geom', 423,
'LINESTRING', 2)

SELECT AddGeometryColumn('geoCamino', 'geom', 423,
'LINESTRING', 2)

- La creación de una tabla espacial se realiza en dos fases:
- Ejemplo
- CREATE TABLE geoCamino (id int4, nom varchar(25))
 - SELECT AddGeometryColumn('public','geoCamino','geom', 423,
'LINESTRING', 2)
- SELECT AddGeometryColumn('geoCamino', 'geom', 423,
'LINESTRING', 2)

La creación de una tabla espacial se realiza en dos fases:

Ejemplo

- CREATE TABLE geoCamino (id int4, nom varchar(25))
- SELECT AddGeometryColumn('public','geoCamino','geom', 423,
'LINESTRING', 2)

SELECT AddGeometryColumn('geoCamino', 'geom', 423,
'LINESTRING', 2)

Base de Datos Espaciales

Creando una tabla espacial

Ejemplos

Se asume que el SRID 128 ya existe

```
CREATE TABLE parque (  
    idParque INTEGER,  
    nombreParque VARCHAR,  
    fechaParque DATE,  
    tipoParque VARCHAR  
);  
  
SELECT AddGeometryColumn('parque', 'geomParque', 128,  
    'MULTIPOLYGON', 2 );
```

Base de Datos Espaciales

Creando una tabla espacial

Ejemplos

Usando una geometría genérica y un SRID sin definir

```
CREATE TABLE caminos (  
  idCamino INTEGER,  
  nombreCamino VARCHAR  
);
```

```
SELECT AddGeometryColumn( 'camino', 'geomCamino', -1,  
  'GEOMETRY', 3 );
```

Base de Datos Espaciales

Cargar datos espaciales en el SDBMS

Actualmente hay dos formas de cargar datos en bases de datos PostGIS / PostgreSQL:

- Usando instrucciones de SQL
- Usando shape loader / dumper

Usando instrucciones SQL

Se puede insertar directamente la información con instrucciones SQL. También se pueden guardar en un archivo texto para que cualquier SDBMS los pueda cargar

Base de Datos Espaciales

Cargar datos espaciales en el SDBMS

Usando instrucciones SQL

Ejemplo de un archivo caminos.sql

```
BEGIN;
INSERT INTO caminos (idCamino,geoCamino,nombreCamino)
VALUES (1,ST_GeomFromText('LINESTRING(191232 243118,191108 243242)',-1),'Av. Caracas');
INSERT INTO caminos (idCamino,geoCamino,nombreCamino)
VALUES (2,ST_GeomFromText('LINESTRING(189141 244158,189265 244817)',-1),'Av. Boyaca');
INSERT INTO caminos (idCamino,geoCamino,nombreCamino)
VALUES (3,ST_GeomFromText('LINESTRING(192783 228138,192612 229814)',-1),'Carrera 30');
INSERT INTO caminos (idCamino,geoCamino,nombreCamino)
VALUES (4,ST_GeomFromText('LINESTRING(189412 252431,189631 259122)',-1),'Av Esperanza');
INSERT INTO caminos (idCamino,geoCamino,nombreCamino)
VALUES (5,ST_GeomFromText('LINESTRING(190131 224148,190871 228134)',-1),'Autopista Sur');
INSERT INTO caminos (idCamino,geoCamino,nombreCamino)
VALUES (6,ST_GeomFromText('LINESTRING(198231 263418,198213 268322)',-1),'Av Centenario');
COMMIT;
```

Estos datos se pueden cargar fácilmente en PotgreSQL usando la instrucción psql:

```
psql -d [database] -f caminos.sql
```

Base de Datos Espaciales

Cargar datos espaciales en el SDBMS

Usando el Cargador (loader).

El cargador de datos “**shp2pgsql**” convierte archivos shape de ESRI a SQL para su inserción en una base de datos PostGIS / PostgreSQL. El cargador tiene varios modos de operación que se seleccionan con los parámetros desde línea de comandos:

- d** Elimina la tabla de la base de datos antes de crear la tabla con los datos del archivo de figuras.
- a** Añade los datos del archivo shape a la tabla de la base de datos. El archivo debe tener los mismos atributo que la tabla.

Base de Datos Espaciales

Cargar datos espaciales en el SDBMS

Usando el Cargador (loader).

- c** Crea una nueva tabla y llena esta con la información del archivo shape. Este es el modo por defecto.
- D** Crea una tabla nueva llenándola con los datos del archivo shape. Pero usa el formato *dump* para la salida de datos que es mas rápido que el *insert* de SQL. Esta forma se debería usar para conjuntos de datos largos.
- s<SRID>** Crea y rellena una tabla geométrica con el SRID que se le pasa como parámetro.

Base de Datos Espaciales

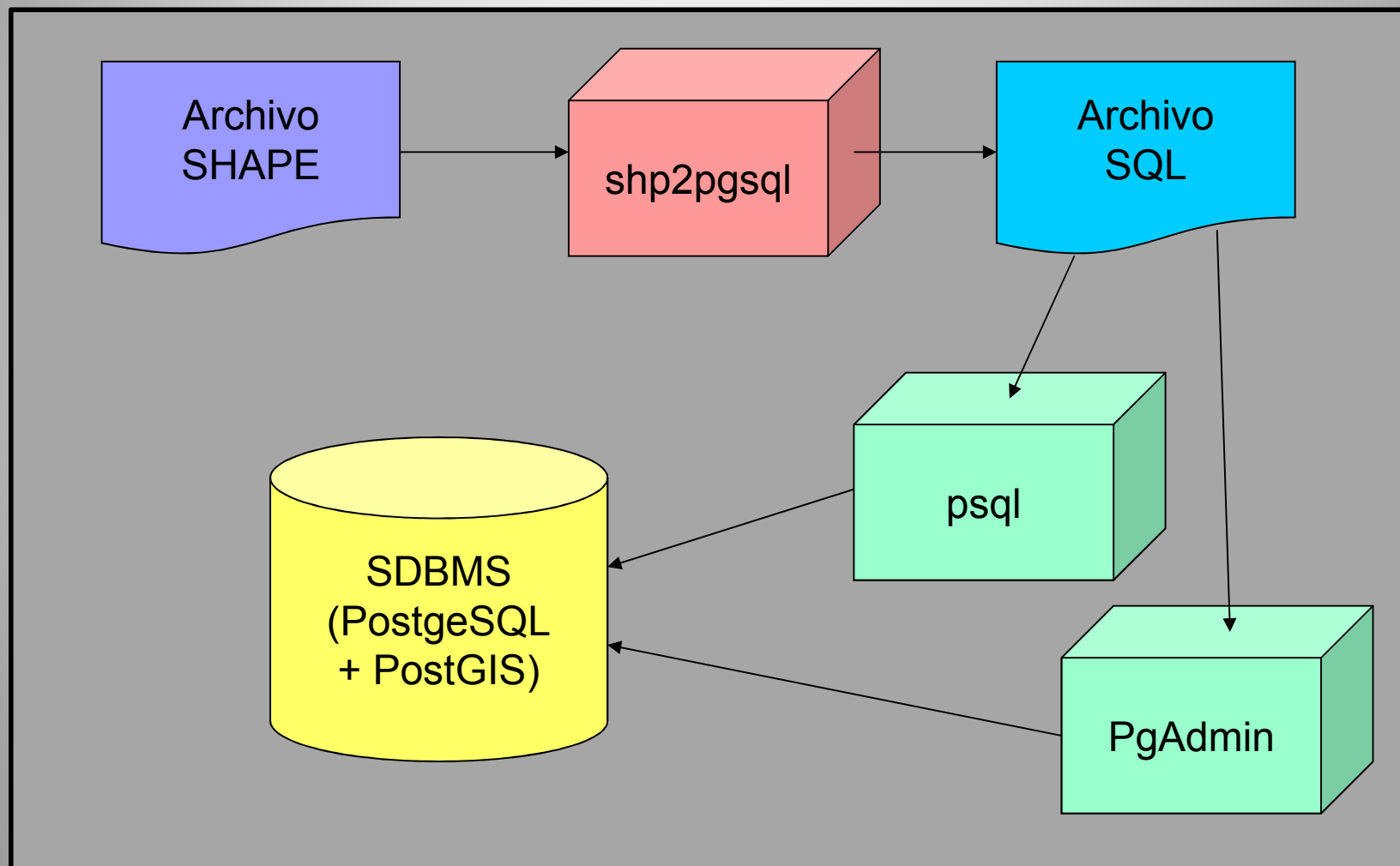
Cargar datos espaciales en el SDBMS

Usando el Cargador (loader).

- p** Solo produce el código SQL de creación de la tabla, sin adicionar dato alguno. Se puede usar cuando se requiere separar la creación de la tabla de la carga de datos en ella.
- I** Crea un índice GiST en la columna geométrica.
- w** Produce una salida en formato WKT, para uso con versiones anteriores de PostGIS.

Base de Datos Espaciales

Cargar datos espaciales en el SDBMS



Base de Datos Espaciales

Recuperando Datos Espaciales

Para extraer datos de la base de datos se pueden utilizar dos métodos:

- Usando instrucciones de SQL
- Usando shape loader / dumper

Usando instrucciones SQL

Es la forma más directa de hacerlo, se realiza una consulta usando la cláusula SELECT de SQL y el resultado se lleva a un archivo plano.

Base de Datos Espaciales

Recuperando Datos Espaciales

Usando instrucciones SQL

```
SELECT idCamino, ST_AsText(geoCamino) AS Geometria, nombreCamino FROM  
caminos;
```

idCamino	Geometria	nombreCamino
----------	-----------	--------------

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

1	LINESTRING(191232 243118,191108 243242)	Av. caracas
---	-----------------------------------------	-------------

2	LINESTRING(189141 244158,189265 244817)	Av. Boyaca
---	-----------------------------------------	------------

3	LINESTRING(192783 228138,192612 229814)	Carrera 30
---	-----------------------------------------	------------

4	LINESTRING(189412 252431,189631 259122)	Av esperanza
---	-----------------------------------------	--------------

5	LINESTRING(190131 224148,190871 228134)	Autopista Sur
---	-----------------------------------------	---------------

6	LINESTRING(198231 263418,198213 268322)	Av Centenario
---	-----------------------------------------	---------------

Base de Datos Espaciales

Recuperando Datos Espaciales

Usando instrucciones SQL

Algunas veces es necesario recortar el numero de registros devueltos. En el caso de restricciones basadas en los atributos usamos la misma sintaxis que para tablas no espaciales. Para restricciones espaciales tenemos los siguientes operadores:

&& Indica cuando la caja que contiene una geometría se superpone a la caja de otra.

~= Me dice si dos geometrías son idénticas. Como

```
'POLIGON((0 0,1 1,1 0,0 0))'~= "POLIGON((0 0,1 1,1 0,0 0))'
```

= Indica si las cajas circunscritas de dos geometrías son iguales.

Base de Datos Espaciales

Recuperando Datos Espaciales

Usando instrucciones SQL

Para usar estos operadores debemos cambiar la representación del formato de texto a geometrías usando la función "GeomFromText()"

```
SELECT idCamino, nombreCamino  
      FROM caminos  
      WHERE geoCamino ~= ST_GeomFromText(  
          'LINESTRING(191232 243118,191108 243242)',-1);
```

Retorna los registros cuya geometría es igual a la dada.

Base de Datos Espaciales

Recuperando Datos Espaciales

Usando instrucciones SQL

Cuando usamos el operador **&&**, se puede especificar como característica de comparación una BOX3D o una GEOMETRY. Cuando se usa ésta última se utiliza una caja como comparación.

```
SELECT idCamino, nombreCamino  
FROM caminos  
WHERE geoCamino && ST_GeomFromText('POLYGON(...))',-1);
```

```
SELECT ST_AsText(geoCamino) AS geometria  
FROM caminos WHERE geoCamino &&  
SetSRID('BOX3D(191232 243117,191232 243119)::box3d,-1);
```

La consulta mas usada es la basada en marcos, que usan los clientes para mostrar un marco del mapa. Podemos usar un BOX3D para el marco. El -1 indica que no se especifica ningún SRID.

Base de Datos Espaciales

Recuperando Datos Espaciales

Usando el cargador (dumper)

Pgsql2shp conecta directamente con la base de datos y convierte una tabla (podría ser un query) en un archivo shape. La sintaxis es:

```
pgsql2shp [<opciones>] <basededatos> <tabla>
```

```
pgsql2shp [<opciones>] <basededatos> <query>
```

Opciones de línea de comandos:

-d Escribe un archivo shape 3D desde una version vieja de PosGIS, siendo el 2D el que tiene por defecto.

-f<archivo> archivo de salida.

-p<puerto> puerto de conexión con la base de datos.

Base de Datos Espaciales

Recuperando Datos Espaciales

Usando el cargador (dumper)

Opciones de línea de comandos:

- h<host>** host donde esta la base de datos.
- p<password>** password para el acceso
- u<user>** usuario de acceso.
- g<columna geometría>** Si la tabla tiene varias columnas geométricas, selecciona la columna geométrica a usar.