

Curso Introducción a SQL Espacial sobre PostGIS

## SQL Espacial

### Funciones constructoras

forma SIG

La plataforma de aprendizaje en SIG Libre



SERVEI DE SISTEMES  
D'INFORMACIÓ GEOGRÀFICA  
I TELEDETECCIÓ  
Universitat de Girona



**UdGFormació**

FUNDACIÓ UNIVERSITAT DE GIRONA:  
INNOVACIÓ I FORMACIÓ

Edita: Servicio de SIG y Teledetección (SIGTE) de la Universitat de Girona

Año: 2014

Contenidos elaborados por: Toni Hernández Vallès

Este documento está sujeto a la licencia Creative Commons BY-NC-SA, es decir, sois libres de copiar, distribuir y comunicar esta obra, bajo las siguientes condiciones:



**Atribución** — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



**No Comercial** — No puede utilizar esta obra para fines comerciales.



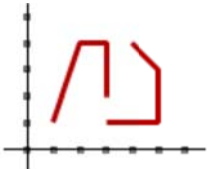
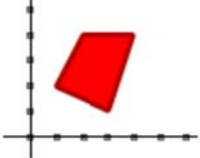
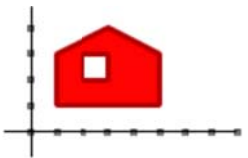
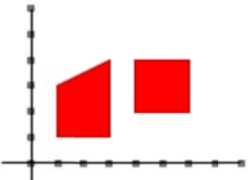

**Compartir bajo la Misma Licencia** — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

## Funciones constructoras

Las funciones espaciales no solo permiten operar con datos espaciales sino que además permiten introducir nuevos datos geométricos en una base de datos espacial. Ahora veremos las principales funciones que nos permitirán introducir nuevos objetos espaciales a partir de datos crudos (indicando las coordenadas de cada nodo). Las funciones generadoras de nuevas geometrías a partir de la transformación de geometrías existentes no se consideran funciones constructoras y serán tratadas en el grupo de funciones que acceden o modifican las propiedades de los objetos geométricos.

Existen dos formatos a partir de los cuales se puede generar objetos espaciales: WKT (Well Known Text) y WKB (Well Known Binary). La siguiente tabla muestra algunos ejemplos de geometrías en formato WKT.

Point	POINT (2 3) 
MultiPoint	MULTIPOINT( 2 3, 4 5, 1 4, 1 1 ) 
LineString	LINESTRING (2 3, 4 5, 4 1, 1 1) 

MultiLinestring	<p>MULTILINESTRING((1 1, 2 4, 3 4, 3 2), (4 4, 5 3, 5 1, 3 1))</p> 
Polygon (sin vacíos)	<p>POLYGON ((3 1, 1 2, 2 4, 4 4, 3 1))</p> 
Polygon (con vacíos)	<p>POLYGON ((1 1, 1 3, 3 4, 5 3, 5 1, 1 1), (2 2, 2 3, 3 3, 3 2, 2 2))</p> 
MultiPolygon	<p>MULTIPOLYGON(((1 1, 1 3, 3 4, 3 1, 1 1)), ((4 2, 4 4, 6 4, 6 2, 4 2)))</p> 
GeometryCollection	<p>GEOMETRYCOLLECTION( RMULTIPOINT(2 4, 6 4), MULTILINESTRING((4 4, 4 2), (1 3, 3 1, 5 1, 7 3)))</p> 

El formato WKB, a diferencia del WKT, no es compresible para los humanos ya que representa las geometrías en formato binario. Cualquier geometría que se puede representar en formato WKT se puede también representar en formato WKB y viceversa.

Veamos ahora cuales son las funciones constructoras de geometrías según el formato de los datos crudos.

## Well Known Text

### **St\_GeomFromText(texto,[srid])**

Permite crear cualquier tipo de geometrías (point, linestring, polygon, etc.) indicando, o no, el sistema de referencia de esas geometrías. Si no se indica el parámetro SRID se asigna el valor por defecto **0**.

Ejemplo:

**INSERT INTO** tabla (id, nombre, the\_geom)

**VALUES** (

1,

'Edificio central',

**St\_GeomFromText('POLYGON((10 28, 12 25, 23 12, 10 28))')**

);

El ejemplo anterior no define el sistema de referencia. Para indicarlo debemos añadir un nuevo parámetro a la función St\_GeomFromText. La nueva expresión quedaría:

**St\_GeomFromText('POLYGON((10 28, 12 25, 23 12, 10 28))', 4326)**

Donde 4326 es el SRID del sistema de coordenadas geográficas.

**St\_PointFromText(texto, [srid])**

Genera una geometría de tipo POINT a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo POINT se produce un error.

Ejemplo

1:

```
SELECT St_PointFromText('POINT(1 2)');
```

Ejemplo 2:

```
SELECT St_PointFromText('POINT(1 2)', 4326);
```

**St\_LineFromText(texto, [srid])**

Genera una geometría de tipo LINESTRING a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo LINESTRING se produce un error.

Ejemplo 1:

```
SELECT St_LineFromText('LINESTRING(1 2, 4 5, 5 6, 7 9)');
```

Ejemplo 2:

```
SELECT St_LineFromText('LINESTRING(1 2, 4 5, 5 6, 7 9)', 4326);
```

**St\_PolyFromText(texto, [srid])****St\_PolygonFromText(texto, [srid])**

Genera una geometría de tipo POLYGON a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo POLYGON se produce un error.

Ejemplo 1:

```
SELECT St_PolyFromText('POLYGON((1 2, 4 5, 5 6, 7 9, 1 2))');
```

Ejemplo 2:

```
SELECT St_PolygonFromText('POLYGON((1 2, 4 5, 5 6, 7 9, 1 2))', 4326);
```

### **St\_MPointFromText(texto, [srid])**

Genera una geometría de tipo MULTIPOINT a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo MULTIPOINT se produce un error.

Ejemplo 1:

```
SELECT St_MPointFromText('MULTIPOINT (1 2, 4 5, 5 6, 7 9, 1 2)');
```

Ejemplo 2:

```
SELECT St_MPointFromText('MULTIPOINT (1 2, 4 5, 5 6, 7 9, 1 2)', 4326);
```

### **St\_MLineFromText(texto, [srid])**

Genera una geometría de tipo MULTILINESTRING a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo MULTILINESTRING se produce un error.

Ejemplo 1:

```
SELECT St_MLineFromText('MULTILINESTRING((1 2, 4 5, 5 6),(7 9, 1 2))');
```

Ejemplo 2:

```
SELECT St_MLineFromText('MULTILINESTRING((1 2, 4 5, 5 6),(7 9, 1 2))', 4326);
```

**St\_MPolyFromText(texto, [srid])**

Genera una geometría de tipo MULTIPOLYGON a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo MULTIPOLYGON se produce un error.

Ejemplo 1:

```
SELECT St_MPolyFromText('MULTIPOLYGON(((1 2, 4 5, 5 6, 1 2)),((7 9, 1 2, 2 4, 7 9)))');
```

Ejemplo 2:

```
SELECT St_MPolyFromText('MULTIPOLYGON(((1 2, 4 5, 5 6, 1 2)),((7 9, 1 2, 2 4, 7 9)))', 4326);
```

**St\_GeomCollFromText(texto, [srid])**

Genera una geometría de tipo GEOMETRYCOLLECTION a partir de la geometría en formato WKT. Si no se indica el parámetro SRID se asigna por defecto **0**. Si la geometría no es de tipo GEOMETRYCOLLECTION se produce un error.

Ejemplo 1:

```
SELECT St_geomCollfromtext('GEOMETRYCOLLECTION(POINT(4 6),LINESTRING(4 6,7 10))');
```

Ejemplo 2:

```
SELECT St_geomCollfromtext('GEOMETRYCOLLECTION(POINT(4 6),LINESTRING(4 6,7 10))',4326);
```

**Well Known Binary**

Como es lógico imaginar, estas funciones están menos extendidas que las funciones que utilizan los datos en formato WKT. Suelen utilizarse para importar datos provenientes de otras



aplicaciones donde los datos están ya almacenados en formato binario. Las siguientes funciones son equivalentes a sus homólogas en formato WKT.

**St\_GeomFromWKB(bytes, [srid])**

**St\_PointFromWKB(texto, [srid])**

**St\_LineFromWKB(texto, [srid])**

**St\_LineStringFromWKB(texto, [srid])**

**St\_PolyFromWKB(texto, [srid])**

**St\_PolygonFromWKB(texto, [srid])**

**St\_MPointFromWKB(texto, [srid])**

**St\_MLineFromWKB(texto, [srid])**

**St\_MPolyFromWKB(texto, [srid])**

**St\_GeomCollFromWKB(texto, [srid])**



SERVEI DE SISTEMES  
D'INFORMACIÓ GEOGRÀFICA  
I TELEDETECCIÓ  
Universitat de Girona



**UdGFormació**

FUNDACIÓ UNIVERSITAT DE GIRONA:  
INNOVACIÓ I FORMACIÓ

[www.sigte.udg.edu/formasig](http://www.sigte.udg.edu/formasig)

[formasig@sigte.org](mailto:formasig@sigte.org)