

Curso Introducción a SQL Espacial sobre PostGIS

SQL Espacial

Funciones de proceso

forma **SIG**

La plataforma de aprendizaje en SIG Libre



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



UdG Formació

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 de Proceso

Dentro de las funciones de proceso se incluyen las funciones que llevan a cabo algún proceso u operación espacial a partir de uno o varios objetos geométricos. Las funciones de proceso más habituales son:

St_Buffer(geometry, radius)

St_Buffer(geometry, radius, num_segments)

St_Buffer(geometry, radius, parameters)

Devuelve la geometría que contiene todos los puntos del sistema de referencia que se encuentran dentro del radio indicado en relación con la geometría inicial.

St_Buffer permite algunas variantes según cual sea el modo de operación. Podemos indicar el número de segmentos utilizados en el redondeo de las geometrías. Por defecto se utilizan 8 segmentos.

Ejemplo:

```
SELECT St_Buffer(
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),1
);
```

Resultado:

Sin especificar (8 Seg)



Ejemplo:

```
SELECT ST_Buffer(
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),1, 1
);
```

Resultado:

1 Segmento



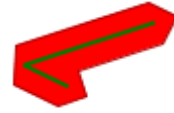
La tercera variante de St_Buffer, permite especificar otros parámetros adicionales. Estos son:

'quad_segs': Número de segmentos utilizados en el redondeo (por defecto 8).

Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'quad_segs=1'  
);
```

Resultado:



'**endcap**': Indica como debe realizarse el buffer en el inicio y final de la geometría. Los valores permitidos son *round* (redondo), *flat* (plano) y *square* (cuadrado).

Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'endcap=round'  
);
```

Resultado:



Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'endcap=flat'  
);
```

Resultado:



Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'endcap=square');
```

Resultado:

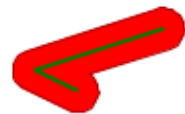


'**join**' : Indica como debe realizarse el buffer en los ángulos internos a la geometría. Los valores permitidos son *round* (redondo), *mitre* (extensión puntiaguda) y *bevel* (sin redondeo ni extensión puntiaguda).

Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'join=round'  
);
```

Resultado:



Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'join=mitre'  
);
```

Resultado:



Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'join=bevel'  
);
```

Resultado:



'mitre_limit': Con este parámetro podemos indicar la longitud máxima de la forma puntiaguda. Solo tiene efecto si se ha especificado el parámetro *join* con el valor *mitre*.

Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'join=mitre mitre_limit=1'  
);
```

Resultado:



Podemos indicar cualquier combinación de parámetros.

Ejemplo:

```
SELECT St_Buffer(  
    St_GeomFromText('linestring(3 1,1 2, 7 4)'),  
    1,  
    'quads_seg=1 endcap=flat join=mitre mitre_limit=1'  
);
```

Resultado:



St_Collect(geometryA, geometryB)

St_Collect(geometry[])

St_Collect une un conjunto de geometrías en una única geometría. A diferencia de la función St_Union que veremos en esta misma lectura, St_Collect no lleva a cabo ninguna operación de disolución de las geometrías implicadas, por lo que resulta bastante más rápida ejecución.

La variante St_Collect(geometry[]) permite recibir una tabla de geometrías.

La geometría resultante puede ser tanto de tipo Multi (*multipoint*, *multilinestring* o *multipolygon*) como GEOMETRYCOLLECTION.

Ejemplo:

```
SELECT St_AsText(St_Collect(
    St_GeomFromText('point(1 1)'),
    St_GeomFromText('point(1 1)'))
);
```

Resultado: "MULTIPOINT(1 1,1 1)".

Como se aprecia en el ejemplo, St_Collect no disuelve las geometrías por lo que las geometrías coincidentes quedan duplicadas.

Si queremos indicar más de dos geometrías deberemos utilizar la función ARRAY[] para convertir las distintas geometrías en una tabla de geometrías. De este modo podemos indicar tantas geometrías como deseemos. Veamos un ejemplo.

Ejemplo:

```
SELECT St_ConvexHull(St_Collect(array [
    St_GeomFromText(LINESTRING(3 1,1 2, 7 4)'),
    St_GeomFromText('MULTIPOINT(1 1, 4 5, -1 2)'),
    St_Buffer(St_GeomFromText('POINT(6 5)'), 1)
]
);
```

Resultado: "GEOMETRYCOLLECTION(LINESTRING(3 1,1 2,7 4),MULTIPOINT(1 1,4 5,-1 2),MULTIPOINT(1 3,6 5,5 2))"

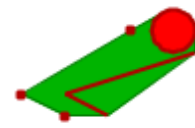
St_ConvexHull(geometry).

Devuelve la mínima geometría que envuelve las geometrías indicadas. Acepta parámetros de tipo MULTI y GEOMETRYCOLLECTION.

Ejemplo:

```
SELECT St_ConvexHull(st_collect(array[
    St_GeomFromText(LINESTRING(3 1,1 2, 7 4)'),
    St_GeomFromText('MULTIPOINT(1 1, 4 5, -1 2)'),
    St_Buffer(St_GeomFromText('POINT(6 5)'), 1)
])
);
```

Resultado:



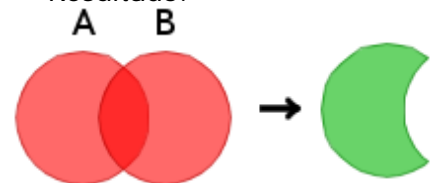
St_Difference(geometryA, geometryB)

Devuelve la porción de geometryA que no intersecciona con geometryB.

Ejemplo:

```
SELECT St_Difference(
    St_Buffer(St_GeomFromText('POINT(5 5)'), 4),
    St_Buffer(St_GeomFromText('POINT(10 5)'), 4)
);
```

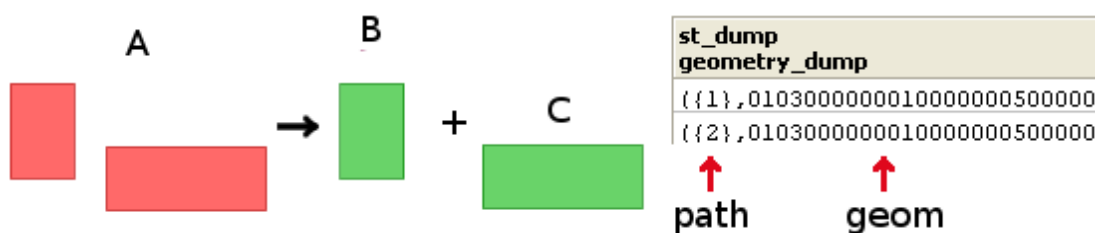
Resultado:



St_Dump(geometry)

Dada una geometría de tipo MULTI (o colección e geometrías) devuelve un conjunto con todas las geometrías NO MULTI que la componen.

Para el siguiente polígono A, st_Dump(geometry) devolverá los polígonos B y C como entidades separadas.



St_Dump no devuelve objetos de geometría de manera directa, sino que devuelve objetos con dos propiedades: path y geom tal y como se muestra en la figura anterior.

La sintaxis correcta para acceder a la parte 'geom' es: `(st_dump(the_geom)).geom`. Si necesitamos hacer un buffer de 2000 metros sobre el resultado de `St_Dump` entonces la sintaxis será (especial atención al uso de paréntesis):

```
select st_buffer((st_dump(the_geom)).geom, 2000) from ...
```

Ejemplo:

```
SELECT St_Dump(  
    St_St_GeomFromText('MULTIPOLYGON(((1 1,1 5,6 5, 6 4,2 4, 2 2, 6 2, 6 1,1  
    1)),((7 6,9 6,9 4,8 4,7 6)))', 4326)  
);
```

Resultado:

	st_dump geometry_dump
1	{{1},0103000020E610000001000000090000000000
2	{{2},0103000020E610000001000000050000000000

St_DumpPoints(geometry)

Devuelve el conjunto de puntos que intervienen en la definición de una geometría. El resultado de la función presenta el mismo formato que la función St_Dump.

Ejemplo:

```
SELECT St_DumpPoints(  
    St_GeomFromText( 'POLYGON((1 1, 1 5, 5 5, 5 1, 1 1),  
    ( 2 2, 2 3, 3 3, 3 2, 2 2))')  
);
```

El ejemplo anterior devuelve un total de 10 puntos (2 repetidos) como geometrías independientes.

Resultado:

st_dumppoints geometry_dump
{{1,1}},01010000000000000000000000000000F03F00
{{1,2}},01010000000000000000000000000000F03F00
{{1,3}},01010000000000000000000000000000144000
{{1,4}},01010000000000000000000000000000144000
{{1,5}},01010000000000000000000000000000F03F00
{{2,1}},010100000000000000000000000000004000
{{2,2}},010100000000000000000000000000004000
{{2,3}},01010000000000000000000000000000084000
{{2,4}},01010000000000000000000000000000084000
{{2,5}},010100000000000000000000000000004000


St_DumpRings(geometry)

Para geometrías de tipo MULTIPOLYGON devuelve todos los anillos (exteriores e interiores) como geometrías de tipo POLYGON. El formato de salida de St_DumpRings es igual al formato de St_Dump.

Ejemplo:

```
SELECT (St_DumpRings(
    St_St_GeomFromText('POLYGON((2 1,2 6, 8 6,8 1,2 1),(3 5,4 5,4 4,3 4, 3 5),
    (6 5,7 5, 7 4, 6 4, 6 5),(4 3, 6 3, 6 2, 4 2, 4 3))', 4326))
);
```

Resultado:



st_dumprings geometry_dump
{{0}},01030000020E61000000
{{1}},01030000020E61000000
{{2}},01030000020E61000000
{{3}},01030000020E61000000

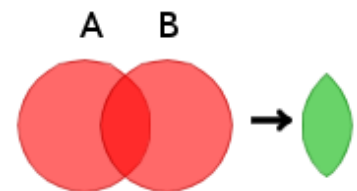
St_Intersection(geometryA, geometryB)

Devuelve una geometría con la superficie compartida por ambas geometrías.

Ejemplo:

```
SELECT St_Intersection(
    St_Buffer(St_GeomFromText('POINT(5 5)'), 4),
    St_Buffer(St_GeomFromText('POINT(10 5)'), 4)
);
```

Resultado:



St_Simplify(geometry, tolerance)

Simplifica geometrías en función de la tolerancia especificada. Una geometría simplificada se convierte en más ligera (para ser transferida por la red) a la vez que pierde precisión. Las unidades del parámetro de tolerancia coinciden con las unidades del sistema de coordenadas del objeto geográfico.

Ejemplo: 1:

```
SELECT St_Simplify(
    St_GeomFromText('POLYGON(( 2 6, 3 7, 4 7, 4 6, 5 6, 5 5, 6 5, 6 4, 7 4, 7 3, 8
    3, 8 2, 9 2, 9 1, 6 1, 6 3, 4 3, 4 5, 2 5, 2 6))')
    ,
    1
);
```

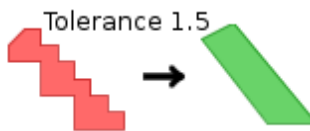
Resultado:



Ejemplo 2:

```
SELECT St_simplify(
    St_GeomFromText('POLYGON(( 2 6, 3 7, 4 7, 4 6, 5 6, 5 5, 6 5, 6 4, 7 4, 7 3, 8
    3, 8 2, 9 2, 9 1, 6 1, 6 3, 4 3, 4 5, 2 5, 2 6))')
    ,
    1.5
);
```

Resultado:



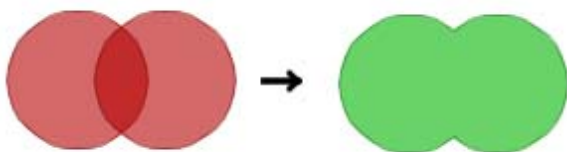
St_Union(geometry)

Esta es una función de agregado como las que hemos visto en la segunda unidad de este curso. Devuelve una sola geometría con la disolución de todos los puntos de la colección de geometrías.

Ejemplo:

```
SELECT St_Union(
    St_Buffer(St_GeomFromText('POINT(5 5)'), 4),
    St_Buffer(St_GeomFromText('POINT(10 5)'), 4)
);
```

Resultado:





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



UdGFormació

FUNDACIÓ UNIVERSITAT DE GIRONA:
INNOVACIÓ I FORMACIÓ

www.sigte.udg.edu/formasig

formasig@sigte.org