

Introducción a SQL espacial con PostGIS

SQL Espacial

Funciones de Medición y relación entre entidades



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

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 Medición y relación entre entidades

En esta lectura hemos incluido dos tipos de funciones distintas: las funciones de medida y las funciones de relación entre geometrías.

Las funciones de medida, realizan algún tipo de medición sobre una geometría dada y devuelven un valor numérico. Las funciones de relación entre entidades evalúan una relación espacial concreta entre dos entidades espaciales dadas y devuelven el resultado de esa evaluación. El resultado puede ser tanto un objeto de tipo geometría como un valor booleano (cierto o falso).

Funciones de medida

St_Area(geometry)

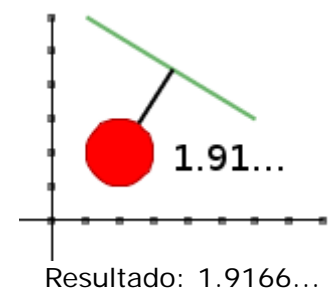
Devuelve la superficie de una geometría de tipo POLYGON o MULTIPOLYGON. El resultado se expresa en las unidades propias del sistema de referencia. Para tipo de datos *geography* el resultado se expresa en metros cuadrados.

St_Distance(geometry, geometry)

Devuelve la mínima distancia entre dos geometrías en un espacio cartesiano de 2 dimensiones. El resultado se expresa en las mismas unidades propias del sistema de referencia.

El siguiente ejemplo muestra en color negro la distancia mínima entre las dos entidades geométricas

Ejemplo:



```
SELECT St_Distance(  
    St_Buffer(St_GeomFromText('POINT(2 2)'),1)
```

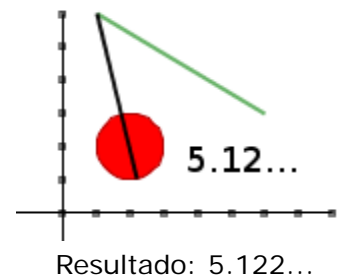
```
,
St_GeomFromText('linestring(1 6, 6 3)')
);
```

St_MaxDistance(geometry, geometry)

Devuelve la distancia máxima entre dos geometrías en un espacio cartesiano de 2 dimensiones.

El resultado se expresa en las mismas unidades del sistema de referencia de las geometrías.

El siguiente ejemplo muestra en color negro la distancia máxima entre las dos entidades geométricas.



Ejemplo:

```
SELECT St_MaxDistance(
    St_Buffer(St_GeomFromText('POINT(2 2)'),1)
    ,
    St_GeomFromText('linestring(1 6, 6 3)')
);
```

St_Distance_Sphere(geometry, geometry)

Devuelve la distancia lineal en metros entre dos geometrías de tipo POINT expresadas en coordenadas geográficas (latitud, longitud). Los cálculos se llevan a cabo sobre una superficie esférica con un radio de 6370986. La unidad del resultado es el metro.

St_Distance_Spheroid(geometry, geometry, spheroid)

Devuelve la distancia lineal en metros entre dos geometrías de tipo POINT expresadas en coordenadas geográficas (latitud, longitud). Los cálculos se llevan a cabo sobre el esferoide indicado. Esta función es más precisa que la anterior pero requiere de cálculos más complejos por lo que el tiempo de respuesta es algo mayor. La unidad del resultado es el metro.

St_Length(geometry)

St_Length2D(geometry)

Devuelve la longitud de una geometría tipo LINESTRING o MULTILINESTRING calculada en un espacio de 2 dimensiones. Las unidades son las propias del sistema de referencia de la geometría.

St_Length3D(geometry)

Devuelve la longitud de una geometría tipo LINESTRING o MULTILINESTRING calculada en un espacio de 3 dimensiones. Si la geometría está en 2 dimensiones, el resultado también se calculará en un espacio de dos dimensiones. Las unidades son las propias del sistema de referencia de la geometría.

St_Length_Spheroid(geometry, spheroid)

St_length3D_Spheroid(geometry, spheroid)

Calcula la longitud de la geometría en un espacio de 2 o 3 dimensiones, dependiendo del número de dimensiones de la geometría, sobre el elipsoide indicado. El resultado se expresa en metros.

St_length2D_Spheroid(geometry, spheroid)

Calcula la longitud de la geometría en dos dimensiones sobre el esferoide indicado. El resultado se expresa en metros.

St_Perimeter(geometry)

St_Perimeter2D(geometry)

Calcula el perímetro de una geometría tipo POLYGON o MULTIPOLYGON sobre un espacio de dos dimensiones. Devuelve 0 si la geometría no contiene área. El resultado se expresa en las unidades propias del sistema de referencia de la geometría.

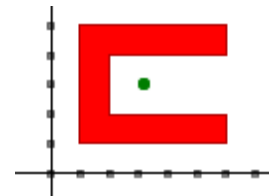
St_Perimeter3D(geometry)

Calcula el perímetro de una geometría de tipo POLYGON o MULTIPOLYGON sobre un espacio de tres dimensiones. Si la geometría indicada está definida en dos dimensiones, entonces calcula el perímetro sobre un espacio de dos dimensiones. El resultado se expresa en las unidades propias del sistema de referencia de la geometría.

Funciones de relación espacial

St_Centroid(centroid)

Devuelve un punto equivalente al centro de masa de un polígono donde todos los puntos tienen la misma masa. Puede devolver puntos que caen fuera de la geometría tal y como se muestra en la siguiente imagen.



Ejemplo:

Resultado:

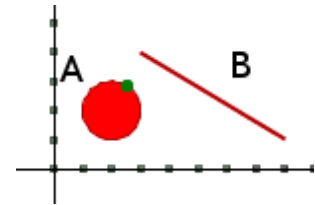
```
SELECT St_Centroid(
    St_GeomFromText(
        'Polygon((1 1,1 5,6 5, 6 4,2 4, 2 2, 6 2, 6 1,1 1))'
    )
);
```

St_Equals(geometry, geometry)

Devuelve cierto si las entidades representan la misma geometría. La dirección de sus vértices no se tiene en cuenta.

St_ClosestPoint(geometryA, geometryB)

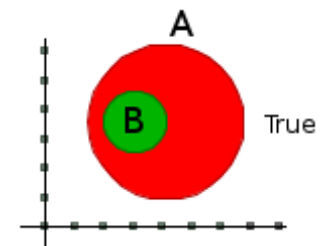
Devuelve un punto, de un espacio de dos dimensiones, perteneciente a la geometríaA más cercano a la geometríaB. La siguiente imagen muestra en verde el punto más cercano entre las dos geometrías.



Ejemplo:

Resultado:

```
SELECT St_AsText (
  St_ClosestPoint(
    St_Buffer(St_GeomFromText('POINT(2 2)'), 1
  ) ,
  St_GeomFromText('LINESTRING(3 4, 8 1)') )
);
```



St_Contains(geometryA, geometryB)

Devuelve cierto únicamente si todos los puntos de la geometryB están dentro de la geometryA.

Ejemplo 1:

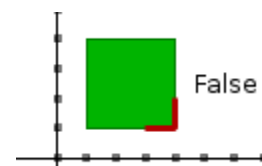
Resultado: True

```
SELECT St_Contains(
  (SELECT St_buffer(geomfromtext('POINT(5 5)'), 5))
,
  (SELECT St_buffer(geomfromtext('POINT(3 5)'), 2))
);
```

Ejemplo 2:

Resultado: False

```
SELECT St_Contains(
```



```
St_GeomFromText('POLYGON((1 1, 1 4, 4 4, 4 1, 1 1))')
,
St_GeomFromText('LINESTRING(3 1, 4 1, 4 2)')
);
```

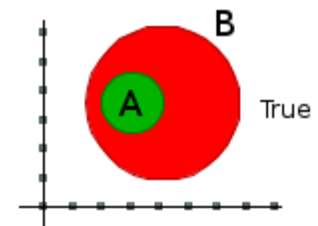
St_Within(geometryA, geometryB)

Devuelve cierto si la geometryA está totalmente incluida en la geometryB. St_Contains(A,B) es equivalente a St_Within(B,A)

Ejemplo:

```
SELECT St_Within(
  (SELECT St_Buffer(St_GeomFromText('POINT(3 5)'), 2)) ,
  (SELECT St_buffer(St_GeomFromText('POINT(5 5)'), 5)));
```

Resultado: True



St_DWithin(geometryA, geometryB)

Devuelve cierto si la distancia mínima entre las geometrías A y B es inferior o igual a 'd'.

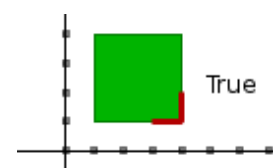
St_Covers(geometryA, geometryB)

Devuelve cierto si no existe ningún punto de geometryB que esté fuera de la geometryA. La única diferencia con St_Contains es que devuelve cierto si la geometría B ocupa solo parte del límite de la geometríaA.

Ejemplo 1:

```
SELECT St_Covers(
  St_GeomFromText('POLYGON((1 1, 1 4, 4 4, 4 1, 1 1))')
,
  St_GeomFromText('LINESTRING(3 1, 4 1, 4 2)')
```

Resultado: True

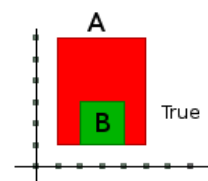


);

Ejemplo 2:

```
SELECT St_Covers(
  St_GeomFromText('POLYGON((1 1, 1 6, 5 6, 5 1, 1 1))')
,
  St_GeomFromText('POLYGON((2 1, 2 3, 4 3, 4 1, 2 1))')
);
```

Resultado: True



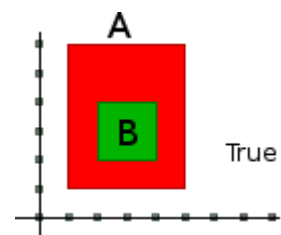
St_ContainsProperly(geometryA, geometryB)

Devuelve cierto si la geometríaB intersecciona con el interior de la geometríaA pero sin interseccionar con el exterior (boundary)

Ejemplo 1:

```
SELECT St_ContainsProperly(
  St_GeomFromText('POLYGON((1 1, 1 6, 5 6, 5 1, 1 1))')
,
  St_GeomFromText('POLYGON((2 2, 2 4, 4 4, 4 2, 2 2))')
);
```

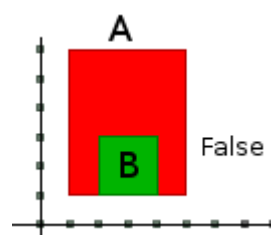
Resultado: True



Ejemplo 2:

```
SELECT St_ContainsProperly(
  St_GeomFromText('POLYGON((2 2, 1 6, 5 6, 5 2, 2 2))')
,
  St_GeomFromText('POLYGON((2 2, 1 6, 5 6, 5 2, 2 2))')
);
```

Resultado: False



```
St_GeomFromText('POLYGON((2 2, 2 4, 4 4, 4 2, 2 2))')
);
```

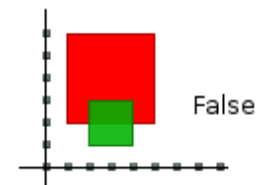
St_Crosses(geometryA, geometryB)

Devuelve cierto si existe algún punto en común, pero no todos, entre ambas geometrías. Además, la intersección entre geometrías debe tener una dimensión menor que la dimensión de las geometrías originales. Es decir, si las geometrías A y B son geometrías de 2 dimensiones (polygon), entonces el resultado de la intersección entre ambas debe tener como máximo una dimensión (point o linestring).

Ejemplo 1:

```
SELECT St_Crosses(
    St_GeomFromText('Polygon((1 2, 1 6, 5 6, 5 2, 1 2))')
,
    St_GeomFromText('Polygon((2 1, 2 3, 4 3, 4 1, 2 1))')
);
```

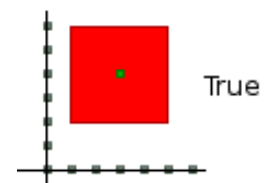
Resultado: False



Ejemplo 2:

```
SELECT St_crosses(
    St_GeomFromText('Polygon((1 2, 1 6, 5 6, 5 2, 1 2))')
,
    St_GeomFromText('Point(3 4)')
);
```

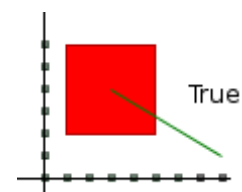
Resultado: True



Ejemplo 3:

```
SELECT St_Crosses(
```

Resultado: True

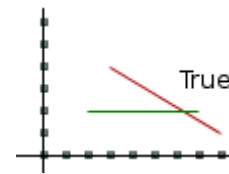


```
St_GeomFromText('Polygon((1 2, 1 6, 5 6, 5 2, 1 2))')
,
St_GeomFromText('Linestring((3 4, 8 1))')
);
```

Ejemplo 4:

```
SELECT St_Crosses(
    St_GeomFromText('Linestring((3 4, 8 1))')
,
    St_GeomFromText('Linestring((2 2, 7 2))')
);
```

Resultado: True



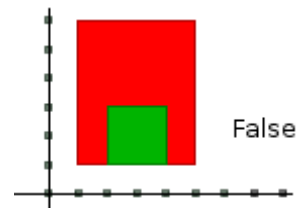
St_Overlaps(geometryA, geometryB)

Devuelve cierto si ambas geometrías comparten algún punto en común y pertenecen a un mismo espacio dimensional pero no están completamente incluidas una en la otra.

Ejemplo 1:

```
SELECT St_Overlaps(
    St_geomFromText('POLYGON((1 1, 1 6, 5 6, 5 1, 1 1))')
,
    St_GeomFromText('POLYGON((2 1, 2 3, 4 3, 4 1, 2 1))')
);
```

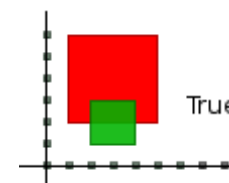
Resultado: False



Ejemplo 2:

```
SELECT St_Overlaps(
    St_GeomFromText('Polygon((1 2, 1 6, 5 6, 5 2, 1 2))')
,
    St_GeomFromText('Polygon((2 1, 2 3, 4 3, 4 1, 2 1))')
);
```

Resultado: True



```
St_GeomFromText('Polygon((2 1, 2 3, 4 3, 4 1, 2 1))')
);
```

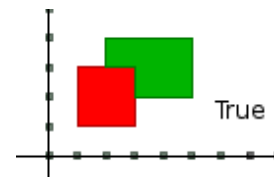
St_Touches(geometryA, geometryB)

Devuelve cierto si existe como mínimo un punto que pertenece a ambas geometrías. Además esos puntos en común no deben pertenecer al interior de las geometrías.

Ejemplo 1:

```
SELECT St_Touches(
    St_GeomFromText('Polygon((1 1, 3 1, 3 3, 1 3, 1 1))')
    ,
    St_GeomFromText('Polygon((2 3, 2 4, 5 4, 5 2, 3 2, 3 3, 2 3))')
);
```

Resultado: True



Ejemplo 2:

```
SELECT St_Touches(
    St_GeomFromText('linestring(4 4, 8 1)')
    ,
    St_GeomFromText('linestring(5 1, 5 6)')
);
```

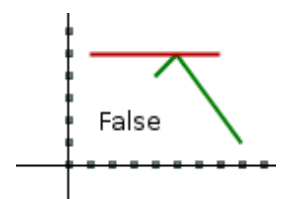
Resultado: False



Ejemplo 3:

```
SELECT St_Touches(
    St_GeomFromText('linestring(4 4, 5 5, 8 1)')
    ,
    St_GeomFromText('linestring(1 5 , 6 5)')
);
```

Resultado: False

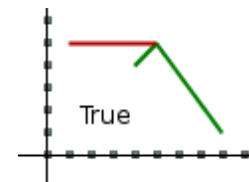


);

Ejemplo 4:

```
SELECT St_Touches(
    St_GeomFromText('linestring(4 4, 5 5, 8 1)')
    ,
    St_GeomFromText('linestring(1 5 , 5 5)')
);
```

Resultado: True



St_Disjoint(geometryA, geometryB)

Devuelve cierto si ambas geometrías no interseccionan.

St_Intersects(geometryA, geometryB)

Devuelve cierto si ambas geometría interseccionan.

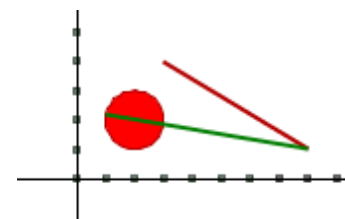
St_LongestLine(geometryA, geometryB)

Devuelve una línea de dos dimensiones que representa la máxima distancia entre ambas geometrías.

Ejemplo:

```
SELECT St_LongestLine(
    St_Buffer(St_GeomFromText('POINT(2 2)'), 1)
    ,
    St_GeomFromText('LINESTRING(3 4, 8 1)')
);
```

Resultado:



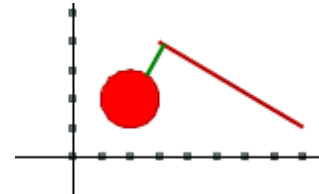
St_ShortestLine(geometryA, geometryB)

Devuelve una línea de dos dimensiones que representa la mínima distancia entre ambas geometrías.

Ejemplo:

```
SELECT St_LongestLine(
  St_Buffer(St_GeomFromText('POINT(2 2)'), 1)
,
  St_GeomFromText('LINESTRING(3 4, 8 1)')
);
```

Resultado:



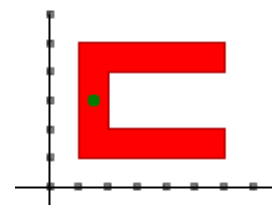
St_PointOnSurface(geometryA, geometryB)

Devuelve una geometría de tipo POINT que cae dentro de la superficie de la geometría inicial. Para un mismo polígono siempre devuelve el mismo valor.

Ejemplo:

```
SELECT St_PointOnSurface(
  St_GeomFromText('Polygon((1 1,1 5,6 5, 6 4,
    2 4, 2 2, 6 2, 6 1,1 1))')
);
```

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