PostgreSQL数据库提供了点、 线、 矩形、 多边形等几何类型， 其他数据库大都没有这些类型。

**几何类型概况**

PostgreSQL主要支持一些二维的几何数据类型。 最基本的类型是“point”， 它是其他类型的基础。

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 存储尺寸 | 表示 | 描述 |
| point | 16字节 | 平面上的点 | (x,y) |
| line | 32字节 | 无限长的线 | {A,B,C} |
| lseg | 32字节 | 有限线段 | ((x1,y1),(x2,y2)) |
| box | 32字节 | 矩形框 | ((x1,y1),(x2,y2)) |
| path | 16+16n字节 | 封闭路径（类似于多边形） | ((x1,y1),...) |
| path | 16+16n字节 | 开放路径 | [(x1,y1),...] |
| polygon | 40+16n字节 | 多边形（类似于封闭路径） | ((x1,y1),...) |
| circle | 24字节 | 圆 | <(x,y),r>（中心点和半径） |

**几何类型的输入**

**点：**

select '1,1'::point;

select '(1,1)'::point;

select point '1,1';

select point '(1,1)';

**线：**

线由线性方程Ax + By + C = 0 表示，其中A和B都不为零。

{ A, B, C }

select '{1,-1,0}'::line;

select '(1,1),(2,2)'::line;

**线段：**

select '1,1,2,2'::lseg;

select '(1,1),(2,2)'::lseg;

select '((1,1),(2,2))'::lseg;

select '[(1,1),(2,2)]'::lseg;

**矩形：**

select box '1,1,2,2';

select box '(1,1),(2,2)';

select box '((1,1),(2,2))';

select box '[(1,1),(2,2)]'; --报错，矩形类型不能使用中括号

**路径：**

select path '1,1,2,2,3,3,4,4';

select path '(1,1),(2,2),(3,3),(4,4)';

select path '((1,1),(2,2),(3,3),(4,4))';

select path '[(1,1),(2,2),(3,3),(4,4)]';

select '1,1,2,2,3,3,4,4'::path;

select '(1,1),(2,2),(3,3),(4,4)'::path;

select '((1,1),(2,2),(3,3),(4,4))'::path;

select '[(1,1),(2,2),(3,3),(4,4)]'::path;

注意：在路径中方括号“[]”表示开放路径， 而圆括号“()”表示闭合路径。 闭合路径是指最后一个点与第一个点是连接在一起的。

**多边形：**

select polygon '1,1,2,2,3,3,4,4';

select polygon '(1,1),(2,2),(3,3),(4,4)';

select polygon '((1,1),(2,2),(3,3),(4,4))';

select polygon '[(1,1),(2,2),(3,3),(4,4)]';--报错，多边形类型中不能使用中括号[]

select '1,1,2,2,3,3,4,4' ::polygon;

select '(1,1),(2,2),(3,3),(4,4)' ::polygon;

select '((1,1),(2,2),(3,3),(4,4))'::polygon;

select polygon '[(1,1),(2,2),(3,3),(4,4)]'; --报错，多边形类型中不能使用中括号[]

**圆型：**

select circle '1,1,5';

select circle '((1,1),5)';

select circle '<(1,1),5>';

select '1,1,5'::circle;

select '((1,1),5)'::circle;

select '<(1,1),5>'::circle;

select circle '(1,1),5';

select '(1,1),5'::circle;

**几何操作符**

“+”、“-”：平移运算符

“\*”“/”：缩放/旋转

**点：**

select point '(1,2)' + point '(10,20)';

select point '(1,2)' - point '(10,20)';

select point '(1,2)' \* point '(10,20)';

select point '(1,2)' / point '(10,20)';

**矩形与点：**

select box '((0,0),(1,1))' + point '(2,2)';

select box '((0,0),(1,1))' - point '(2,2)';

**路径与点：**

select path '(0,0),(1,1),(2,2)' + point '(10,20)';

select path '(0,0),(1,1),(2,2)' - point '(10,20)';

**圆与点：**

select circle '((0,0),1)' + point '10,20';

select circle '((0,0),1)' - point '10,20';

乘数 point ‘x,0’， 相当于几何对象缩放x倍：

select point '(1,2)' \* point '(2,0)';

select point '(1,2)' \* point '(3,0)';

select circle '((0,0),1)' \* point '(3,0)';

select circle '((1,1),1)' \* point '(3,0)';

乘数 point ‘0,1’， 相当于几何对象逆时针旋转90度， 而 point ‘0,-1’， 则表示顺时针旋转90度：

select point '(1,2)' \* point '(0,1)';

select point '(1,2)' \* point '(0,-1)';

select circle '((0,0),1)' \* point '(0,1)';

select circle '((1,1),1)' \* point '(0,1)';

**几何运算符**

**运算符“#”**

1.对于两个线段， 计算出交点。

2.对于两个矩形， 计算出相交的矩形。

3.对于路径或多边形， 则计算出顶点数。

两个线段：

select lseg '(0,0), (2,2)' # lseg '(0,2), (2,0)'; --返回焦点

select lseg '(0,0), (1,1)' # lseg '(2,2), (3,3)';--返回空

两个矩形：

select box '(0,0), (2,2)' # box '(1,0), (3,1)';

路径或多边形：

select # path '(1,1), (2,2), (3,3)';

select # polygon '(1,1), (2,2), (3,3)';

**运算符“@-@”**

运算符“@-@” 参数的类型只能为“lseg”“path”。 一般用于计算几何对象的长度。

计算线段长度：

select @-@ lseg '(0,0), (1,1)';

计算path长度：

select @-@ path '(0,0), (2,2)';

select @-@ path '(0,0), (1,1),(2,2)';

注意， 开放式路径与闭合路径的长度是不一样的：

select @-@ path '[(0,0), (1,1),(0,1)]';

select @-@ path '(0,0), (1,1),(0,1)';

**运算符“@@”**

运算符“@@”为一元运算符， 用于计算中心点， 示例如下：

select @@ circle '<(1,1), 2>';

select @@ box '(0,0), (1,1)' ;

select @@ lseg ‘(0,0), (1,1)';

**运算符“##”**

运算符“##”为二元运算符， 用于计算两个几何对象上距离最近的点， 示例如下：

select point '(0,0)' ## lseg '((2,0),(0,2))';

select point '(0,0)' ## box '((1,1),(2,2))';

select lseg '(1,0),(0,1.5)' ## lseg '((2,0),(0,2))';

**运算符“<->”**

运算符“<->”为二元运算符， 用于计算两个几何对象之间的间距， 示例如下：

select lseg '(0,1),(1,0)' <-> lseg '((0,2),(2,0))';

select circle '((0,0),1)' <-> circle '((3,0),1)';

select circle '((0,0),1)' <-> circle '((2,2),1)';

对于两个矩形来说， 它们之间的间距实际上是中心点之间的距离：

select box '((0,0),(1,1))' <-> box '((2,0),(4,1))';

select box '((0,0),(1,1))' <-> box '((1,1),(2,2))';

**运算符“&&”**

运算符“&&” 用于计算两个几何对象之间是否重叠， 只要有一个共同点， 返回结果即为真， 示例如下：

select box '((0,0),(1,1))' && box '((0,0),(1,1))';

select box '((0,0),(1,1))' && box '((0,0),(1,1))';

select circle '((0,0),1)' && circle '((1,1),1)';

select circle '((0,0),1)' && circle '((2,2),1)';

select polygon '(0,0),(2,2),(0,2)' && polygon '(0,1),(1,1),(2,0)';

**判断两个对象相对位置的运算符**

判断左右位置的运算符有4个， 具体如下。

>>： 严格在右。circle '<(5,0),1>' >> circle '<(0,0),1>' → t

&<： 延展到左。box '(1,1),(0,0)' &< box '(2,2),(0,0)' → t

&>： 延展到右。 box '(3,3),(0,0)' &> box '(2,2),(0,0)' → t

判断上下位置的运算符有6个， 具体如下：

<<|： 严格在下。box '(3,3),(0,0)' <<| box '(5,5),(3,4)' → t

|>>： 严格在上。box '(5,5),(3,4)' |>> box '(3,3),(0,0)' → t

&<|： 延展到下。box '(1,1),(0,0)' &<| box '(2,2),(0,0)' → t

|&>： 延展到上。box '(3,3),(0,0)' |&> box '(2,2),(0,0)' → t

<^： 在下面（允许接触） 。

point '(1,0)' <^ point '(1,1)' → t

box '((1,1),(0,0))' <^ box '((2,2),(1,1))' → t

>^： 在上面（允许接触） 。

point '(1,1)' >^ point '(1,0)' → t

box '((2,2),(1,1))' >^ box '((1,1),(0,0))' → t

判断两个对象相对位置的其他运算符如下：

?#： 是否相交。lseg '[(-1,0),(1,0)]' ?# box '(2,2),(-2,-2)' → t

?-： 是否水平或水平对齐。

?- lseg '[(-1,0),(1,0)]' → t

point '(1,0)' ?- point '(0,0)' → t

?|： 是否竖直或竖直对齐。

?| lseg '[(-1,0),(1,0)]' → f

point '(0,1)' ?| point '(0,0)' → t

?-|： 两个对象是否垂直。lseg '[(0,0),(0,1)]' ?-| lseg '[(0,0),(1,0)]' → t

?||： 两个对象是否平行。lseg '[(-1,0),(1,0)]' ?|| lseg '[(-1,2),(1,2)]' → t

@>： 是否包含。circle '<(0,0),2>' @> point '(1,1)' → t

<@： 包含或在其上。point '(1,1)' <@ circle '<(0,0),2>' → t

判断两个几何对象是否相同的运算符“~=”

对于多边形， 如果表示的起点不同， 但实现上它们是两个相同的多边形， 那么相应的判断代码如下：

select polygon '((0,0),(1,1))' ~= polygon '((1,1),(0,0))';--t

select polygon '((0,0),(1,1),(1,0))' ~= polygon '((1,1),(0,0),(1,0))';--t

对于矩形， 示例代码如下：

select box '(0,0),(1,1)' ~= box '(1,1),(0,0)';--t

**几何函数**

|  |  |  |
| --- | --- | --- |
| 函数 | 描述 | 例子 |
| area | 计算面积 | area(box '(2,2),(0,0)') → 4 |
| center | 计算中心点 | center(box '(1,2),(0,0)') → (0.5,1) |
| diameter | 计算圆的直径 | diameter(circle '<(0,0),2>') → 4 |

**几何类型转换函数**

|  |  |  |
| --- | --- | --- |
| 函数 | 描述 | 例子 |
| box ( circle ) → box | 将圆转换成矩形 | box(circle '<(0,0),2>') |
| box ( point ) → box | 将点转换为空框 | box(point '(1,0)') → (1,0),(1,0) |
| box ( point, point ) → box | 将任意两个角点转换为框。 | box(point '(0,1)', point '(1,0)') → (1,1),(0,0) |
| circle ( point, double precision ) → circle | 将圆心和半径转换成圆 | circle(point '(0,0)', 2.0) → <(0,0),2> |
| line ( point, point ) → line | 将两个点转换成通过它们的直线。 | line(point '(-1,0)', point '(1,0)') → {0,-1,0} |
| point ( circle ) → point | 计算圆心 | point(circle '<(0,0),2>') → (0,0) |

其他函数：http://www.postgres.cn/docs/14/functions-geometry.html