

mysql存储与计算地理位置信息

冒险小A [关注](#)

2019.09.25 18:48:57 字数 1,931 阅读 1,725

mysql存储字段类型：

- Geometry

Geometry是几何对象的基类, 也就是说Point, LineString, Polygon都是Geometry的子类,

- Point

点对象, 有一个坐标值, 没有长度、面积、边界。

数据格式为『经度(longitude)在前, 纬度(latitude)在后, 用空格分隔』 例: POINT(121.213342 31.234532)

- LineString

线对象, 由一系列点连接而成。

如果线从头至尾没有交叉, 那就是简单的 (simple)

如果起点和终点重叠, 那就是封闭的 (closed)

数据格式为『点与点之间用逗号分隔; 一个点中的经纬度用空格分隔, 与POINT格式一致』 例: LINESTRING(121.342423 31.542423,121.345664 31.246790,121.453178 31.456862)

- Polygon

多边形对象。可以是一个实心平面形, 即没有内部边界, 也可以有空洞, 类似纽扣

数据格式为

『实心型: 一个表示外部边界的LineString和0个表示内部边界的LineString组成』 例:

POLYGON((121.342423 31.542423,121.345664 31.246790,121.453178 31.456862),(121.563633 31.566652,121.233565 31.234565,121.568756 31.454367))

『纽扣型: 一个表示外部边界的LineString和多个表示内部边界的LineString组成』 例: POLYGON((0 0,10 0, 10 10, 0 10))

- MultiPoint, MultiLineString, MultiPolygon, GeometryCollection

为以上对象的集合。

数据格式为下例

MULTIPOINT(0 0, 20 20, 60 60)

MULTILINESTRING((10 10, 20 20), (15 15, 30 15))

MULTIPOLYGON(((0 0,10 0,10 10,0 10,0 0)),((5 5,7 5,7 7,5 5)))

GEOMETRYCOLLECTION(POINT(10 10), POINT(30 30), LINESTRING(15 15, 20 20))

常用函数：

一、几何对象属性查询函数:

Geometry(为基类函数, 点线面都可用)

1. Dimension(g) : 返回几何对象g的维数, 点为0, 线为1, 多边形为2
2. Envelope(g): 返回几何对象g的最小边界矩形(xy的极值点)。如果对象为点则返回该点对象, 如果对象为线和多边形则返回极值xy坐标构造的矩形Polygon
3. GeometryType(g): 返回几何对象g的类型名称, 点为POINT, 线为LINEPOINT, 多边形为POLYGON
4. IsClosed(g): 返回几何对象g是否封闭, 条件为该线对象首尾point重合则为封闭, 封闭为1, 不封闭为0, 如果几何对象不为线对象的话, 返回为null

5. IsSimple(g): 返回几何对象g是否简单, 条件为该线对象路径没有交叉则为简单, 简单为1, 不简单为0, 如果几何对象不为线对象的话, 返回为null

Point

- 1. X(p): 返回该点X坐标
- 2. Y(p): 返回改点Y坐标

LineString

- 1. EndPoint(line): 返回对象line的最后一个点Point
- 2. StartPoint(line): 返回对象line的第一个点Point
- 3. PointN(line, N): 返回对象line中第N个点, N从1开始

Polygon

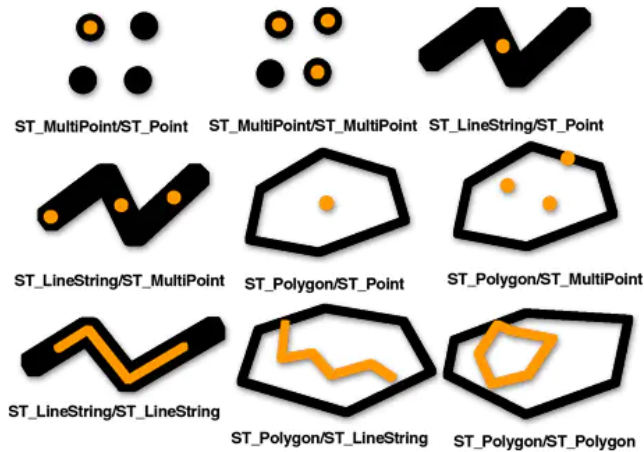
- 1. ExteriorRing(poly): 返回多边形对象poly的外轮廓线, 类型为LineString
- 2. InteriorRingN(poly, N): 返回对象poly的第N个空洞轮廓线, N从1开始
- 3. NumInteriorRings(poly): 返回对象poly的空洞个数

二、返回新的几何对象

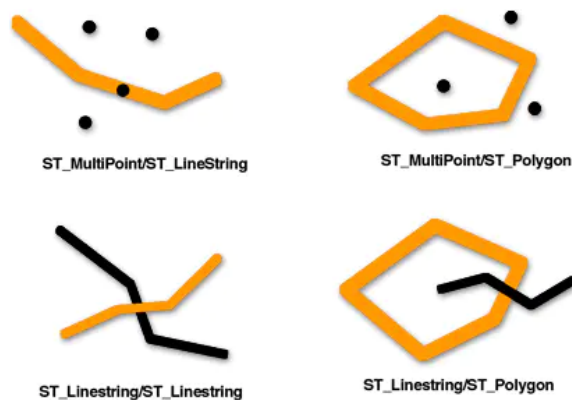
- 1. st_union(g1, g2): 返回 面1和面2的并集
- 2. st_difference(g1, g2): 返回 面1 - (面1和面2的交集)
- 3. st_intersection(g1, g2): 返回 面1和面2的交集

三、查询几何对象关系

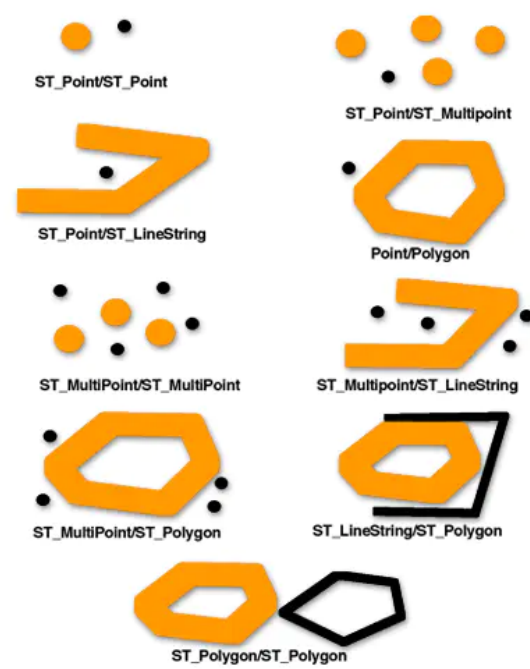
- 1. ST_Contains(a,b): 如果几何对象a完全包含几何对象b, 则返回1, 否则0



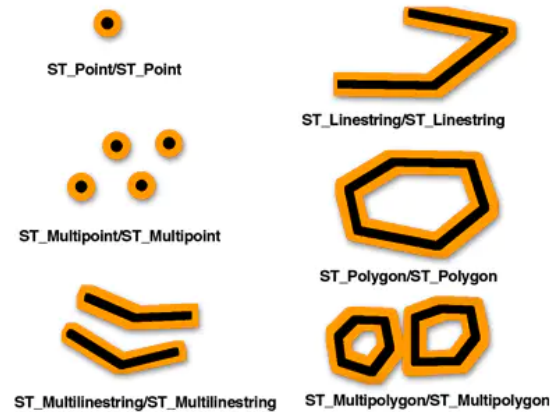
- 2. ST_Crosses(a,b): 如果a横跨b,则返回1,否则返回0



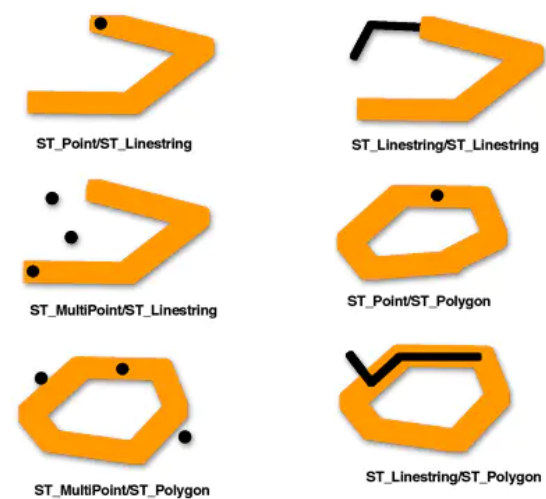
3. `ST_Disjoint(a,b)`: 如果a和b不相交,则返回1.否则返回0



4. `ST_Equals(a,b)`: 如果a和b有相同的几何描述,则返回1, 否则返回0; 例如一栋楼的两层xy坐标描述一致,所以返回为1



- 5. `ST_Intersects(a,b)`: 与`ST_Disjoint`结果完全相反
- 6. `ST_Overlaps(a,b)`: 两个维度相同的几何对象相交的交集是一样维度的几何对象时, 返回1, 否则返回0
- 7. `ST_Touches(a,b)`: 几何对象a交且只交于b的边界时, 返回1, 否则0



8. `ST_Within(a,b)`: 与`ST_Contains(a,b)`结果完全相反

四、数值计算

1. `ST_Distance_Sphere(POINT(30 40), POINT(20 30))`:计算两点之间距离(千米)

五、描述语言转化成几何对象

1. `geomfromtext()`: 空间函数中, 参数不可直接写空间描述格式, 需要用`geomfromtext()`来将描述语言转化成函数的对象.例如, 要查找test表中, 所有和 `POLYGON((4 4, 4 6, 6 6, 6 4,4 4))` 相交的多边形, 则sql写为 `select polygon1 from test where st_disjoint(geomfromtext('POLYGON((4 4, 4 6, 6 6, 6 4,4 4))'),polygon1) = 0`

六、附：MySQL空间相关函数一览表

The following table lists each spatial function and provides a short description of each one.
Name | Description

- 1. `Area()` Return Polygon or MultiPolygon area
- 2. `AsBinary()`, `AsWKB()` Convert from internal geometry format to WKB
- 3. `AsText()`, `AsWKT()` Convert from internal geometry format to WKT
- 4. `Buffer()` Return geometry of points within given distance from geometry
- 5. `Centroid()` Return centroid as a point
- 6. `Contains()` Whether MBR of one geometry contains MBR of another
- 7. `Crosses()` Whether one geometry crosses another
- 8. `Dimension()` Dimension of geometry
- 9. `Disjoint()` Whether MBRs of two geometries are disjoint
- 10. `EndPoint()` End Point of LineString
- 11. `Envelope()` Return MBR of geometry
- 12. `Equals()` Whether MBRs of two geometries are equal
- 13. `ExteriorRing()` Return exterior ring of Polygon
- 14. `GeomCollFromText()`,`GeometryCollectionFromText()` Return geometry collection from WKT
- 15. `GeomCollFromWKB()`,`GeometryCollectionFromWKB()` Return geometry collection from WKB
- 16. `GeometryCollection()` Construct geometry collection from geometries
- 17. `GeometryN()` Return N-th geometry from geometry collection
- 18. `GeometryType()` Return name of geometry type
- 19. `GeomFromText()`,`GeometryFromText()` Return geometry from WKT
- 20. `GeomFromWKB()`,`GeometryFromWKB()` Return geometry from WKB
- 21. `GLength()` Return length of LineString
- 22. `InteriorRingN()` Return N-th interior ring of Polygon
- 23. `Intersects()` Whether MBRs of two geometries intersect
- 24. `IsClosed()` Whether a geometry is closed and simple
- 25. `IsEmpty()` Placeholder_function
- 26. `IsSimple()` Whether a geometry is simple
- 27. `LineFromText()`,`LineStringFromText()` Construct LineString from WKT
- 28. `LineFromWKB()`,`LineStringFromWKB()` Construct LineString from WKB
- 29. `LineString()` Construct LineString from Point values
- 30. `MBRContains()` Whether MBR of one geometry contains MBR of another
- 31. `MBRDisjoint()` Whether MBRs of two geometries are disjoint
- 32. `MBREqual()` Whether MBRs of two geometries are equal
- 33. `MBRIntersects()` Whether MBRs of two geometries intersect

34. `MBROverlaps()` Whether MBRs of two geometries overlap
35. `MBRTouches()` Whether MBRs of two geometries touch
36. `MBRWithin()` Whether MBR of one geometry is within MBR of another
37. `MLineFromText()`, `MultiLineStringFromText()` Construct `MultiLineString` from WKT
38. `MLineFromWKB()`, `MultiLineStringFromWKB()` Construct `MultiLineString` from WKB
39. `MPointFromText()`, `MultiPointFromText()` Construct `MultiPoint` from WKT
40. `MPointFromWKB()`, `MultiPointFromWKB()` Construct `MultiPoint` from WKB
41. `MPolyFromText()`, `MultiPolygonFromText()` Construct `MultiPolygon` from WKT
42. `MPolyFromWKB()`, `MultiPolygonFromWKB()` Construct `MultiPolygon` from WKB
43. `MultiLineString()` Construct `MultiLineString` from `LineString` values
44. `MultiPoint()` Construct `MultiPoint` from `Point` values
45. `MultiPolygon()` Construct `MultiPolygon` from `Polygon` values
46. `NumGeometries()` Return number of geometries in geometry collection
47. `NumInteriorRings()` Return number of interior rings in `Polygon`
48. `NumPoints()` Return number of points in `LineString`
49. `Overlaps()` Whether MBRs of two geometries overlap
50. `Point()` Construct `Point` from coordinates
51. `PointFromText()` Construct `Point` from WKT
52. `PointFromWKB()` Construct `Point` from WKB
53. `PointN()` Return N-th point from `LineString`
54. `PolyFromText()`, `PolygonFromText()` Construct `Polygon` from WKT
55. `PolyFromWKB()`, `PolygonFromWKB()` Construct `Polygon` from WKB
56. `Polygon()` Construct `Polygon` from `LineString` arguments
57. `SRID()` Return spatial reference system ID for geometry
58. `ST_Area()` Return `Polygon` or `MultiPolygon` area
59. `ST_AsBinary()`, `ST_AsWKB()` Convert from internal geometry format to WKB
60. `ST_AsText()`, `ST_AsWKT()` Convert from internal geometry format to WKT
61. `ST_Buffer()` Return geometry of points within given distance from geometry
62. `ST_Centroid()` Return centroid as a point
63. `ST_Contains()` Whether one geometry contains another
64. `ST_Crosses()` Whether one geometry crosses another
65. `ST_Difference()` Return point set difference of two geometries
66. `ST_Dimension()` Dimension of geometry
67. `ST_Disjoint()` Whether one geometry is disjoint from another
68. `ST_Distance()` The distance of one geometry from another
69. `ST_EndPoint()` End Point of `LineString`
70. `ST_Envelope()` Return MBR of geometry
71. `ST_Equals()` Whether one geometry is equal to another
72. `ST_ExteriorRing()` Return exterior ring of `Polygon`
73. `ST_GeomCollFromText()`, `ST_GeometryCollectionFromText()`, `ST_GeomCollFromText()` Return geometry collection from WKT
74. `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()` Return geometry collection from WKB
75. `ST_GeometryN()` Return N-th geometry from geometry collection
76. `ST_GeometryType()` Return name of geometry type
77. `ST_GeomFromText()`, `ST_GeometryFromText()` Return geometry from WKT
78. `ST_GeomFromWKB()`, `ST_GeometryFromWKB()` Return geometry from WKB
79. `ST_InteriorRingN()` Return N-th interior ring of `Polygon`
80. `ST_Intersection()` Return point set intersection of two geometries


- 81. `ST_Intersects()` Whether one geometry intersects another
- 82. `ST_IsClosed()` Whether a geometry is closed and simple
- 83. `ST_IsEmpty()` Placeholder_function
- 84. `ST_IsSimple()` Whether a geometry is simple
- 85. `ST_LineFromText()`,`ST_LineStringFromText()` Construct LineString from WKT
- 86. `ST_LineFromWKB()`,`ST_LineStringFromWKB()` Construct LineString from WKB
- 87. `ST_NumGeometries()` Return number of geometries in geometry collection
- 88. `ST_NumInteriorRing()`,`ST_NumInteriorRings()` Return number of interior rings in Polygon
- 89. `ST_NumPoints()` Return number of points in LineString
- 90. `ST_Overlaps()` Whether one geometry overlaps another
- 91. `ST_PointFromText()` Construct Point from WKT
- 92. `ST_PointFromWKB()` Construct Point from WKB
- 93. `ST_PointN()` Return N-th point from LineString
- 94. `ST_PolyFromText()`,`ST_PolygonFromText()` Construct Polygon from WKT
- 95. `ST_PolyFromWKB()`,`ST_PolygonFromWKB()` Construct Polygon from WKB
- 96. `ST_SRID()` Return spatial reference system ID for geometry
- 97. `ST_StartPoint()` Start Point of LineString
- 98. `ST_SymDifference()` Return point set symmetric difference of two geometries
- 99. `ST_Touches()` Whether one geometry touches another
- 100. `ST_Union()` Return point set union of two geometries
- 101. `ST_Within()` Whether one geometry is within another
- 102. `ST_X()` Return X coordinate of Point
- 103. `ST_Y()` Return Y coordinate of Point
- 104. `StartPoint()` Start Point of LineString
- 105. `Touches()` Whether one geometry touches another
- 106. `Within()` Whether MBR of one geometry is within MBR of another
- 107. `X()` Return X coordinate of Point
- 108. `Y()` Return Y coordinate of Point



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下




冒险小A 真正的大师，永远都怀着一颗学徒的心。

文章基本是自己的学习笔记和理解，非正确参考...

总资产48 共写了3.1W字 获得36个赞 共19个粉丝

关注



写下你的评论...