### 1.空间类型概述

MySQL空间扩展支持地理特征的生成，存储和分析：

* 用于表示空间值的数据类型
* 用于处理空间值的函数
* 空间索引提高了空间列访问时间

一些空间数据类型保存单个几何值：

* GEOMETRY 几何
* POINT 点
* LINESTRING 线
* POLYGON 多边形

GEOMETRY可以存储任何类型的几何值。其他单值类型（POINT， LINESTRING和POLYGON）将其值限制为特定的几何类型

其他空间数据类型保存值的集合：

* MULTIPOINT 点集
* MULTILINESTRING 多个线集
* MULTIPOLYGON 多边形集
* GEOMETRYCOLLECTION

GEOMETRYCOLLECTION可以存储任何类型的对象的集合。其他集合类型（MULTIPOINT， MULTILINESTRING， MULTIPOLYGON，和 GEOMETRYCOLLECTION）限制集合成员向那些具有特定的几何形状的类型。

创建一个名为geom具有名称g可以存储任何几何类型值的列的表，请使用以下语句：

CREATE TABLE geom (g GEOMETRY);

### 2.支持的空间数据格式

1.文本（WKT）格式以ASCII形式交换几何数据。

Point POINT(15 20)

LineString（有四个点） LINESTRING(0 0, 10 10, 20 25, 50 60)

Polygon（两个圈起来的环） POLYGON((0 0,10 0,10 10,0 10,0 0),(5 5,7 5,7 7,5 7, 5 5))

 MultiPoint（三个点） MULTIPOINT(0 0, 20 20, 60 60)

MultiLineString有两个 LineString值：MULTILINESTRING((10 10, 20 20), (15 15, 30 15))

  MultiPolygon有两个 Polygon值：MULTIPOLYGON(((0 0,10 0,10 10,0 10,0 0)),((5 5,7 5,7 7,5 7, 5 5)))

GeometryCollection由两个的 Point值和一 LineString：GeometryCollection由两个的 Point值和一 LineString：GEOMETRYCOLLECTION(POINT(10 10), POINT(30 30), LINESTRING(15 15, 20 20))

2. 二进制（WKB）格式几何值的公知二进制（WKB）表示用于交换几何数据作为由[BLOB](https://dev.mysql.com/doc/refman/5.7/en/blob.html)包含几何WKB信息的值表示的二进制流。

### 3.几何形成的良好和有效性

对于几何值，MySQL区分句法良好和几何有效的概念。

如果一个几何体满足这个（非穷举）列表中的条件，那么这个几何体在语法上就是格式良好的：

* 线条至少有两点
* 多边形至少有一个环
* 多边形环关闭（第一个和最后一个点相同）
* 多边形环至少有4个点（最小多边形是第一个和最后一个点相同的三角形）
* 集合不是空的（除了 GeometryCollection）

如果一个几何在句法上是良构的并且满足这个（非穷举）列表中的条件，那么这个几何在几何上是有效的：

* 多边形不是自相交的
* 多边形内圈在外圈内
* 多多边形不具有重叠的多边形

如果几何在语法上不是格式良好的，空间函数就会失败。分析WKT或WKB值的空间导入函数会导致尝试创建非句法结构几何的错误。语法格式良好性也被检查，以尝试将几何存储到表中。

允许插入，选择和更新几何无效的几何图形，但它们必须在语法上格式良好。由于计算开销，MySQL不会明确检查几何有效性。空间计算可能检测到一些无效几何的情况并产生错误，但是它们也可能返回未定义的结果而不检测无效。需要几何形状有效的几何图形的应用程序应该使用[ST\_IsValid()](https://dev.mysql.com/doc/refman/5.7/en/spatial-convenience-functions.html#function_st-isvalid)函数检查它们 。

### 4.创建空间列

MySQL提供了一种为几何类型创建空间列的标准方法，

* 使用该[CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html) 语句来创建具有空间列的表格：

CREATE TABLE geom (g GEOMETRY);

* 使用该[ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html)语句在现有表中添加或删除空间列：
* ALTER TABLE geom ADD pt POINT;

ALTER TABLE geom DROP pt;

### 5.填充空间列

创建空间列后，可以使用空间数据填充它们。

值应以内部几何格式存储，但您可以将其转换为来自已知文本（WKT）或已知二进制（WKB）格式的格式。以下示例演示了如何通过将WKT值转换为内部几何格式将几何值插入表中：

* 在[INSERT](https://dev.mysql.com/doc/refman/5.7/en/insert.html" \o "13.2.5 INSERT语法)声明中直接执行转换 ：
* INSERT INTO geom VALUES (ST\_GeomFromText('POINT(1 1)'));
* SET @g = 'POINT(1 1)';

INSERT INTO geom VALUES (ST\_GeomFromText(@g));

* 执行以下操作之前的转换 [INSERT](https://dev.mysql.com/doc/refman/5.7/en/insert.html)：
* SET @g = ST\_GeomFromText('POINT(1 1)');

INSERT INTO geom VALUES (@g);

以下示例将更复杂的几何图形插入表中：

SET @g = 'LINESTRING(0 0,1 1,2 2)';

INSERT INTO geom VALUES (ST\_GeomFromText(@g));

SET @g = 'POLYGON((0 0,10 0,10 10,0 10,0 0),(5 5,7 5,7 7,5 7, 5 5))';

INSERT INTO geom VALUES (ST\_GeomFromText(@g));

SET @g =

'GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(0 0,1 1,2 2,3 3,4 4))';

INSERT INTO geom VALUES (ST\_GeomFromText(@g));

前面的例子 [ST\_GeomFromText()](https://dev.mysql.com/doc/refman/5.7/en/gis-wkt-functions.html" \l "function_st-geomfromtext)用来创建几何值。您也可以使用类型特定的功能：

SET @g = 'POINT(1 1)';

INSERT INTO geom VALUES (ST\_PointFromText(@g));

SET @g = 'LINESTRING(0 0,1 1,2 2)';

INSERT INTO geom VALUES (ST\_LineStringFromText(@g));

SET @g = 'POLYGON((0 0,10 0,10 10,0 10,0 0),(5 5,7 5,7 7,5 7, 5 5))';

INSERT INTO geom VALUES (ST\_PolygonFromText(@g));

SET @g =

'GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(0 0,1 1,2 2,3 3,4 4))';

INSERT INTO geom VALUES (ST\_GeomCollFromText(@g));

* POINT(1 1)用十六进制文字语法 插入一个值：
* INSERT INTO geom VALUES

(ST\_GeomFromWKB(X'0101000000000000000000F03F000000000000F03F'));

* 在一个C程序中，你可以使用二进制值进行转义 [mysql\_real\_escape\_string\_quote()](https://dev.mysql.com/doc/refman/5.7/en/mysql-real-escape-string-quote.html" \o "27.8.7.56 mysql_real_escape_string_quote（）) ，并将结果包含在发送给服务器的查询字符串中。

### 6.获取空间数据

存储在表格中的几何值可以以内部格式获取。您也可以将它们转换为WKT或WKB格式。

* 以内部格式获取空间数据：

使用内部格式提取几何值可用于表到表传输：

CREATE TABLE geom2 (g GEOMETRY) SELECT g FROM geom;

* 以WKT格式提取空间数据：

该[ST\_AsText()](https://dev.mysql.com/doc/refman/5.7/en/gis-format-conversion-functions.html#function_st-astext)函数将内部格式的几何转换为WKT字符串。

SELECT ST\_AsText(g) FROM geom;

* 以WKB格式获取空间数据：

该[ST\_AsBinary()](https://dev.mysql.com/doc/refman/5.7/en/gis-format-conversion-functions.html#function_st-asbinary)功能将几何从内部格式转换为 [BLOB](https://dev.mysql.com/doc/refman/5.7/en/blob.html)包含WKB值。

SELECT ST\_AsBinary(g) FROM geom;

### 7.创建空间索引（SPATIAL空间索引关键字）

对于[MyISAM](https://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html)和 InnoDB表，可以使用SPATIAL索引来优化包含空间数据的列中的搜索操作 。最典型的操作是：

* 点搜索包含给定点的所有对象的查询
* 搜索与给定区域重叠的所有对象的区域查询

MySQL使用***R-树与二次拆分***为SPATIAL对空间列的索引。

SPATIAL指数使用几何的最小外接矩形（MBR）构建的。对于大多数几何图形，MBR是包围几何图形的最小矩形。对于水平或垂直线串，MBR是退化为线串的矩形。对于某一点，MBR是一个退化到点的矩形。

对于InnoDB和MyISAM 表，MySQL可以使用类似于创建常规索引的语法创建空间索引，但使用 SPATIAL关键字。空间索引中的列必须声明NOT NULL。以下示例演示如何

* 带有[CREATE TABLE](https://dev.mysql.com/doc/refman/5.7/en/create-table.html)：

CREATE TABLE geom (g GEOMETRY NOT NULL, SPATIAL INDEX(g));

* 带有[ALTER TABLE](https://dev.mysql.com/doc/refman/5.7/en/alter-table.html)：
* CREATE TABLE geom (g GEOMETRY NOT NULL);

ALTER TABLE geom ADD SPATIAL INDEX(g);

* 带有[CREATE INDEX](https://dev.mysql.com/doc/refman/5.7/en/create-index.html)：
* CREATE TABLE geom (g GEOMETRY NOT NULL);

CREATE SPATIAL INDEX g ON geom (g);

SPATIAL INDEX创建一个R-tree索引。对于支持空间列非空间索引的存储引擎，引擎会创建一个B树索引。空间值的B树索引对于精确值查找非常有用，但对于范围扫描则不适用

## 8.使用空间索引

优化程序将调查是否可用的空间索引可能涉及搜索使用函数（如[MBRContains()](https://dev.mysql.com/doc/refman/5.7/en/spatial-relation-functions-mbr.html#function_mbrcontains)或 [MBRWithin()](https://dev.mysql.com/doc/refman/5.7/en/spatial-relation-functions-mbr.html" \l "function_mbrwithin)在 WHERE子句中）的查询。以下查询查找给定矩形中的所有对象：mysql> SET @poly =

-> 'Polygon((30000 15000,

31000 15000,

31000 16000,

30000 16000,

30000 15000))';

mysql> SELECT fid,ST\_AsText(g) FROM geom WHERE

-> MBRContains(ST\_GeomFromText(@poly),g);

执行[SELECT](https://dev.mysql.com/doc/refman/5.7/en/select.html" \o "13.2.9 SELECT语法)没有空间索引的语句会得到相同的结果，但会导致执行时间从0.00秒上升到0.46秒：

mysql> SET @poly =

-> 'Polygon((30000 15000,

31000 15000,

31000 16000,

30000 16000,

30000 15000))';

mysql> SELECT fid,ST\_AsText(g) FROM geom IGNORE INDEX (g) WHERE

-> MBRContains(ST\_GeomFromText(@poly),g);