**参考：<http://www.gaia-gis.it/spatialite-2.4.0-4/spatialite-cookbook/index.html#family>**

## 实例一：Splite转Spatialite，创建点要素列

至于[如何从sqlite转换成为spatialite](https://link.jianshu.com?t=http://gis.stackexchange.com/questions/19421/how-do-i-add-and-map-a-non-spatial-table-to-qgis-from-sqlite" \t "_blank)，这是一个非常严肃的问题。很多数据都是从sqlite来的。当然如果你的数据库创建直接在spatialite上，一切都很好说。从sqlite中导入数据到spalite需要一个过程，首先输入数据库命令。

select InitSpatialMetaData();

初始化空间数据库，如果你的数据库容量大，这个过程有点漫长。然后输入命令。

select AddGeometryColumn('你的表名', '你的几何列名',4326, 'POINT');

最后，当然是从你以前的xy中赋值。

update 你的表名 set 你的几何列名 = GeomFromText('POINT ('+纬度列=' '+经度列+')',4326);

也可以这样。

update 你的表名 set 你的几何列名 = MakePoint(X列,Y列,EPSG投影码);

作者：brandonxiang  
链接：https://www.jianshu.com/p/f94d8eecc6b3  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

**注意**：From后面直接跟表名查询结果即就是表的乘积基础上查询，如果想查单个可以把其中一个表图形先进行融合（A\*1=A）再进行判断。

# 判断点是否在线内查询相交：

**select** **asText**(geometry) **from** point **where**

**ST\_Intersects**(point.geometry,**GeomFromText**('POLYGON((322.372427 113.231539, 230.763178 -34.295044, 49.924143 -20.018278, -141.622469 0.20714, -208.247377 109.662348, 83.236597 259.568392, 86.805788 398.766859, 216.486414 426.130659, 310.475122 334.52141, 322.372427 113.231539))'))=1

# 找出A和B相交的部分

**select** **asText**(A.geometry) **from** A,B **where**

**ST\_Intersects**(A.geometry,B.geometry)=1

# A和B相交为空则返回1，反之返回0

**select** **ST\_Disjoint(**A.geometry,B.geometry**)** **from** A,B

# 返回A和B的距离集合

**select** \*,**ST\_Distance(**A.geometry,B.geometry**)** **from** A,B

# 擦除（从A中擦除B）

**select** **asWKT**(**ST\_Difference**(A.geometry,B.geometry)) **from** A,B

**where** **ST\_Difference**(A.geometry,B.geometry) **is** **not** **null**

# 将A中所有要素融合成一个

**select** **asWKT**(**ST\_Union**(A.geometry)) **from** A

# 将A和B融合成一个形状

**select** **asWKT**(**ST\_Union**(point.geometry,polygon.geometry)) **from** point,polygon

# 定义A以3.6为缓冲单位的缓冲区（3.6为对应其空间参考的距离单位）

**select** **asText**(**ST\_Buffer**(A.geometry,3.6)) **from** A

# 判断A是不是简单的

**select** **IsSimple**(A.geometry) **from** A

# 返回了A边界上的所有点集合LineString

**select** **AsText(Boundary**(polygon.geometry)**)** **from** polygon

# 返回A上所有的顶点和对应的基本点

**select** **AsText**(**ST\_DissolvePoints**(A.geometry)) **from** A

# 返回A的面积、返回A的质心（可能在面外）

**select** **ST\_Area**(A.geometry) **from** A

**select** **AsText(ST\_Centroid**(A.geometry)**)** **from** A

# 获取点的XYZM参数

**select** **ST\_X**(point.geometry) **as** X,**ST\_Y**(point.geometry) **as** Y,**ST\_Z**(point.geometry) **as** Z,**ST\_M**(point.geometry) **as** M **from** point

# 获取其对应的SRID参考

**select** **ST\_SRID**(geometry) **from** ..TableName..

# 创建空间表

（1）首先创建一个对应不含空间信息列的空表

**create** **table** ..TableName..(

ID **integer** **primary** **key** **autoincrement**,

Name **text** **not** **null**)

（2）增加存储空间信息的列

**select** **AddGeometryColumn**('..TableName..','Geometry',4326,'Point','XY');

（表名，列名字，srid，数据类型，坐标类型）

（3）增加其空间索引

**select** **CreateSpatialIndex**('..TableName..','Geometry');

（4）插入空间数据(需要填写与之对应的SRID)

**insert** **into** ..TableName.. **values**(2,0,**geomfromtext**('POINT(-111.8720226.255935)',4326));

**insert** **into** ..TableName.. **values**(2,0,**MakePoint**(-111.879206,226.255935,4326));

从其它表查询获取获取数据

**insert** **into** pr (geometry) **select** geometry **from** prt