

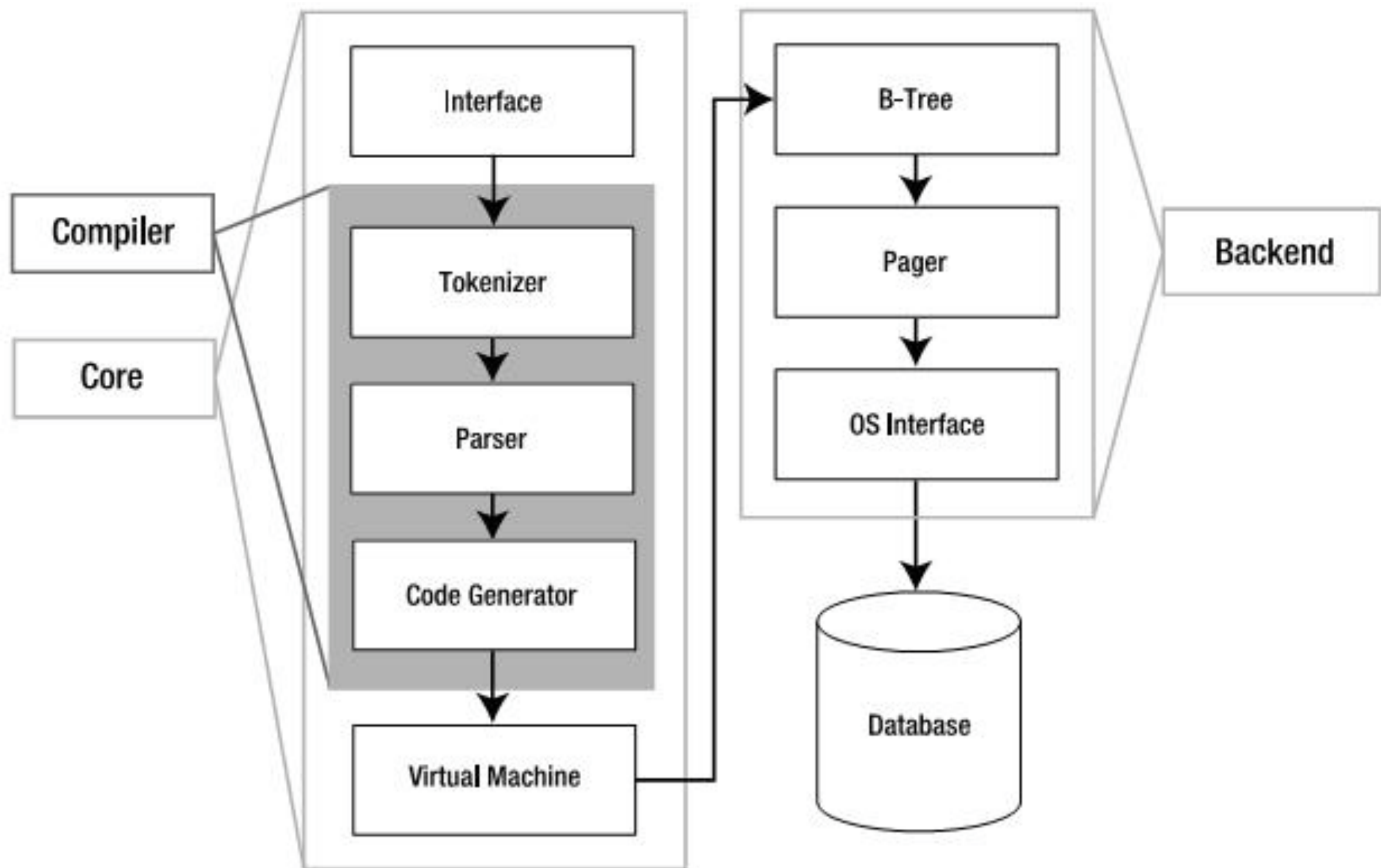
SQLite
&
SQLite Spatial
vs
PostGIS

胡林舒

SQLite 简介

- 关系型数据库：遵守ACID
- 轻型数据库：
 - 包含在一个相对小的C库中
 - 设计目标是嵌入式，在嵌入式设备中占有资源非常低，只需要几百k内存即可
- 跨平台：能够支持Windows/Linux/Unix等
- 多语言程序：支持Tcl、C#、PHP、Java等，还有ODBC接口
- 计算速度快：相比于MySQL和PostgreSQL这两款同样的开源数据库，都相对更快一点

SQLite 架构



SQLite 特性

1. ACID事务
2. 零配置 – 无需安装和管理配置
3. 储存在单一磁盘文件中的一个完整的数据库
4. 数据库文件可以在不同字节顺序的机器间自由的共享
5. 支持数据库大小至2TB
6. 足够小, 大致13万行C代码, 4.43M
7. 比一些流行的数据库在大部分普通数据库操作要快
8. 简单, 轻松的API
9. 包含TCL绑定, 同时通过Wrapper支持其他语言的绑定
10. 良好注释的源代码, 并且有着90%以上的测试覆盖率
11. 独立: 没有额外依赖
12. 源码完全的开源, 你可以用于任何用途, 包括出售它
13. 支持多种开发语言, C, C++, PHP, Perl, Java, C#,Python, Ruby等

SQLite的数据类型

- 特色: Typelessness
- 常见类型: varchar, nvarchar, text, integer, float, boolean, clob, blob, timestamp, numeric, varying character, national varying character

SQLite 客户端可视化工具

- SQLiteMan, 使用QT开发的一个SQLite客户端, 支持多语言、跨平台。
- SQLite Manager, 以火狐浏览器的扩展形式提供的SQLite客户端。
- SQLite Database Browser
- SqlPro SQL Client

SpatialLite

● 简单介绍

- 使用C++库开发

- 定义：SQLite的空间扩展

- 功能：支持SQLite数据库按照OGC的Simple Feature Access标准存取空间数据并进行一些简单的数据处理和运算

SpatiaLite的基本几何类型

- Point/MultiPoint
- LineString/MultiLineString
- Polygon/MultiPolygon
- Curve/MultiCurve
- Surface/MultiSurface
- GeometryCollection

SpatiaLite的主要功能

- 支持WKT和WKB格式
- 支持SQL空间函数AsText(), GeomFromText(), Area(), PointN()
- 对OpenGIS空间函数集的完全支持需要借助于GEOS, 支持高级复杂空间分析函数如: Overlaps(), Touches(), Union(), Buffer()等
- 遵循OpenGIS规范完全支持空间元数据格式
- 支持对Shape文件的导入与导出
- 采用PROJ.4和EPSG支持坐标系的投影变换
- 基于SQLite的RTree扩展实现了空间索引
- VirtualShape扩展使得SQLite访问Shape文件就像操作VIRTUAL TABLE一样
- 用户可以对外部Shape文件进行标准SQL查询操作, 而无需导入或者转换Shape文件
- VirtualText扩展使得SQLite访问CSV/TxtTab文件就像操作VIRTUAL TABLE一样
- 用于可以对外部CSV/TxtTab文件进行标准SQL查询操作, 而无需导入或者转换CSV/TxtTab文件

Spatialite 最新版本添加的功能

- added **math** functions
- added **EXIF-GPS geometry-handling** functions
- added the new Virtual Table module **VirtualNetwork**, supporting **routing** [Dijkstra's Shortest Path **algorithm**], 类似于pgrouting, 但是功能尚不完全, 或者说, 还很简陋, 大家有兴趣研究其用法可以看: <http://blog.3snews.net/space.php?uid=6188&do=blog&id=64331>
- added the new Virtual Table module **VirtualFDO** supporting **compatibility** (兼容性) for FDO/OGR RFC16

Spatialite支持的几何数据函数

(简单举例)

MakePoint(x Double precision , y Double precision , [, SRID Integer]) : Geometry — ST_MakePoint (ST_Point)

BuildMbr(x1 Double precision , y1 Double precision , x2 Double precision , y2 Double precision [, SRID Integer]) : Geometry
— ST_MakeBox2D

BuildCircleMbr(x Double precision , y Double precision , radius Double precision [, SRID Integer]) : Geometry
— ST_Point_Inside_Circle

MbrMinX(geom Geometry) : Double precision (also Y & Max)
— ST_XMin

GeomFromText(wkt String [, SRID Integer]) : Geometry (also Point, LineString, Polygon, Multi* and GeomCollection & WKB)
— ST_GeomFromText

AsText(geom Geometry) : String(WKT) — ST_AsText

AsBinary(geom Geometry) : Binary(WKB) — ST_AsBinary

Dimension(geom Geometry) : Integer — ST_Dimension
GeometryType(geom Geometry) : String — ST_GeometryType
SRID(geom Geometry) : Integer — ST_SRID
SetSRID(geom Geometry , SRID Integer) : Integer — ST_SetSRID
IsEmpty(geom Geometry) : Integer — ST_IsEmpty
IsSimple(geom Geometry) : Integer — ST_IsSimple
IsValid(geom Geometry) : Integer — ST_IsValid
Boundary(geom Geometry) : Geometry — ST_Boundary
Envelope(geom Geometry) : Geometry — ST_Envelope
X(pt Point) : Double precision — ST_X
Y(pt Point) : Double precision — ST_Y
IsClosed(c Curve) : Integer — ST_IsClosed
IsRing(c Curve) : Integer — ST_IsRing
NumPoints(Line LineString) : Integer — ST_NumPoints
PointN(Line LineString , n Integer) : Point — ST_PointN
Centroid(s Surface) : Point — ST_Centroid
Area(s Surface) : Double precision — ST_Area

Disjoint(geom1 Geometry , geom2 Geometry) : Integer — ST_Disjoint
Overlaps(geom1 Geometry , geom2 Geometry) : Integer — ST_Overlaps
Crosses(geom1 Geometry , geom2 Geometry) : Integer — ST_Crosses
Intersects(geom1 Geometry , geom2 Geometry) : Integer — ST_Intersects
Contains(geom1 Geometry , geom2 Geometry) : Integer — ST_Contains
Distance(geom1 Geometry , geom2 Geometry) : Double precision
— ST_Distance
Buffer(geom Geometry , dist Double precision) : Geometry — ST_Buffer
Intersection(geom1 Geometry , geom2 Geometry) : Geometry
— ST_Intersection
SymDifference(geom1 Geometry , geom2 Geometry) : Geometry
— ST_SymDifference
ConvexHull(geom Geometry) : Geometry
— ST_ConvexHull
AddGeometryColumn(table String , column String , srid Integer ,
geom_type String , dimension Integer) : Integer — AddGeometryColumn
CreateSpatialIndex(table String , column String) : Integer
— CreateIndex... On... Using Gist

SpatialLite 比较特殊的函数

GeomFromFGF(fgfGeometry Binary [, SRID Integer]) : Geometry

AsFGF(geom Geometry) : Binary

SQL functions that test approximative spatial relationships via MBRs: MbrEqual, MbrDisjoint, MbrTouches etc.

SQL functions implementing FDO/OGR compatibility: AutoFDOStart, AutoFDOStop, InitFDOSpatialMetaData, AddFDOGeometryColumn etc. (基于VirtualFDO)

http://www.gaia-gis.it/spatialite-2.3/spatialite-2.3_sql.html

SpatialLite的VirtualShape

(VirtualText 类似)

功能：可以在不将Shapefile文件导入数据库的情况下在原shapefile文件上直接执行Select语句的查询

实现机制：基于VirtualTable将其导入数据库的过程隐藏

SpatialLite的空间索引的实现

使用Virtual Table实现

1. an RTree actually requires three distinct ordinary tables, applying the following naming schema:

- the `indexname_node` table is used to store binary data corresponding to each tree node
- the `indexname_parent` table is used to represent the tree nodes hierarchy
- the `indexname_rowid` table is used to associate each tree node with corresponding ROWIDs

you access the `indexname VIRTUAL TABLE`, containing the following columns:

- the `pkid` column represents the ROWID of the corresponding row within the table associated with this Spatial Index
- the `xmin`, `xmax`, `ymin` and `ymax` columns represents the MBR aka BBOX for corresponding geometry

2. SpatialLite为了确保进入到主表中和主表关联的空间索引表的控件对象的高度一致建立了三个triggers(出发器), 使得每次从主表中INSERT、UPDATE、DELETE一行数据你可以保证相应的索引信息也已经正确的更新了

SpatialLite的空间索引

删除:

```
SELECT DisableSpatialIndex('NewTowns', 'geom');  
DROP TABLE idx_NewTowns_geom;  
VACUUM;
```

1、调用 **DisableSpatialIndex** 这样来移除 trigger，来停止空间索引的更新。

2、因为第一步并不会删除 RTree 自身，所以要显示的调用 **DROP TABLE** 以在物理层面上把索引表删除掉。

3、最好调用一下 **VACUUM**，用以维护数据库的高效状态。

(经过这三步的操作只是把索引和出发器删除了，而元数据表 **geometry_column** 中的相应的数据却没有删除掉，如果要删除的话需要显示的调用 **SQL** 语句来删除这些数据)

另一种实现方式 **based on MBRs in-memory caching**，其实类似，只是信息储存的地方不同，具体大家可以参看 http://www.gaia-gis.it/spatialite-2.3/spatialite-2.3_manual.html (**Memory Buffer Register** 存储器缓冲寄存器)

SQLite的VirtualTable机制

- One cannot create a trigger on a virtual table.
- One cannot create additional indices on a virtual table. (Virtual tables can have indices but that must be built into the virtual table implementation. Indices cannot be added separately using CREATE INDEX statements.)
- One cannot run ALTER TABLE ... ADD COLUMN commands against a virtual table.



Load Shapefile

Path: /Users/mac/Documents/GIS/地理空间数据库/挑战3/Gec

Table name: Abila

GeomColumn name: Geometry

SRID: 4326

Charset Encoding

SHIFT_JIS	Japanese
TCVN	Vietnamese
UTF-8	UNICODE/Universal
VISCII	Vietnamese

OK Cancel

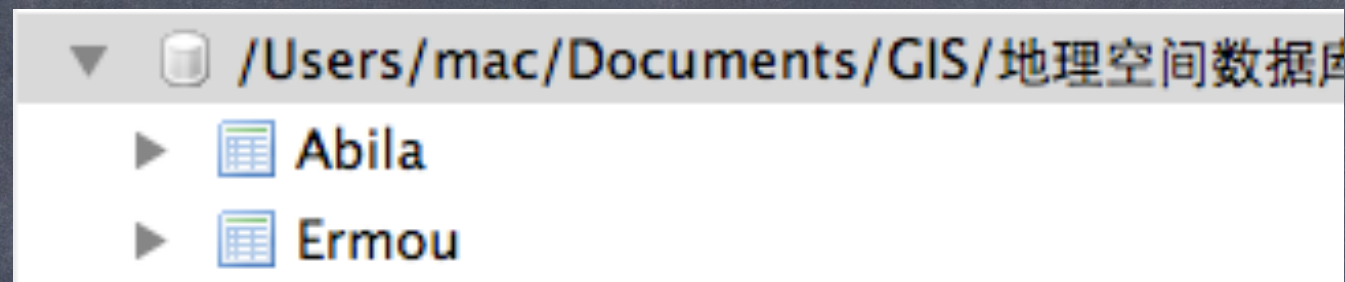
PRAGMA

PRAGMA table_info(sqlite_stat1)

	cid	name	type	notnull	dflt_value	pk
1	0	tbl		0	NULL	0
2	1	idx		0	NULL	0
3	2	stat		0	NULL	0

CREATE

Create Table Ermou (name,geometry);



INSERT

```
insert into Ermou select fename,  
convexhull(geometry) from Abila where  
fename = 'Ermou' group by fename;
```

	name	geometry
1	Ermou	BLOB sz=80 GEOMETRY

VirtualShape

```
create virtual table test_shape  
using virtualshape('/Users/mac/  
Documents/GIS/地理空间数据库/挑战3/  
Geospatial/Abila', UTF-8, 4326);
```



test_shape

	cid	name	type	notnull	dflt_value	pk
1	0	PKUID	INTEGER	0	NULL	0
2	1	Geometry	BLOB	0	NULL	0
3	2	TLID	INTEGER	0	NULL	0
4	3	FEDIRP	VARCHAR(2)	0	NULL	0
5	4	FENAME	VARCHAR(16)	0	NULL	0
6	5	FETYPE	VARCHAR(5)	0	NULL	0
7	6	FEDIRS	VARCHAR(2)	0	NULL	0
8	7	FRADDL	INTEGER	0	NULL	0
9	8	TOADDL	INTEGER	0	NULL	0
10	9	FRADDR	INTEGER	0	NULL	0
11	10	TOADDR	INTEGER	0	NULL	0