

GBASE

空间数据库技术白皮书



GBase 8t 空间数据库技术白皮书，南大通用数据技术股份有限公司

GBase 版权所有©2004-2017，保留所有权利。

版权声明

本文档所涉及的软件著作权、版权和知识产权已依法进行了相关注册、登记，由南大通用数据技术股份有限公司合法拥有，受《中华人民共和国著作权法》、《计算机软件保护条例》、《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的南大通用公司的版权信息由南大通用公司合法拥有，受法律的保护，南大通用公司对本文档可能涉及到的非南大通用公司的信息不承担任何责任。在法律允许的范围内，您可以查阅，并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经南大通用公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将视为侵权，南大通用公司具有依法追究其责任的权利。

本文档中包含的信息如有更新，恕不另行通知。您对本文档的任何问题，可直接向南大通用数据技术股份有限公司告知或查询。

未经本公司明确授予的任何权利均予保留。

通讯方式

南大通用数据技术股份有限公司

天津华苑产业区海泰发展六道 6 号海泰绿色产业基地 J 座(300384)

电话：400-817-9696

邮箱：info@gbase.cn

商标声明

GBase 是南大通用数据技术股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由南大通用公司合法拥有，受法律保护。未经南大通用公司书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯南大通用公司商标权的，南大通用公司将依法追究其法律责任。

目录

1	空间数据概述	1
1.1	地理坐标系.....	1
1.2	投影坐标系.....	2
2	空间数据解决方案.....	2
2.1	应用程序连接空间数据库.....	2
2.2	空间数据复制.....	3
2.3	Web 要素服务.....	3
3	空间数据库	3
3.1	spatial_references 表.....	3
3.2	st_units_of_measure 表.....	4
3.3	geometry_columns 表.....	5
3.4	空间数据表.....	5
4	空间数据类型	6
4.1	数据类型属性.....	6
4.2	ST_Point.....	7
4.3	ST_LineString.....	7
4.4	ST_Polygon.....	8
4.5	ST_MultiPoint.....	9
4.6	ST_MultiLineString.....	9
4.7	ST_MultiPolygon.....	10
5	数据转换	11
5.1	Well-known text (WKT)	11
5.2	Well-known binary (WKB)	11
5.3	ESRI shape(SHP).....	12
5.4	Geography Markup Language (GML).....	12
5.5	Keyhole Markup Language (KML).....	13
6	R 树索引.....	13
6.1	R 树索引结构.....	13
6.2	创建和使用 R 树索引.....	14
7	并行查询	15
8	估算存储空间	16
9	空间数据加载与迁移.....	17
10	空间函数	17

1 空间数据概述

GBase 8t 数据库系统嵌入了地理信息系统 (GIS)，实现地理数据及非地理数据的 SQL 查询。

GBase 8t 的空间数据类型实现了 OpenGIS 联盟(开放地理信息系统, OGC) 抽象数据类型 (ADT) 的 SQL3 规范。这些数据类型可以存储空间数据，如一个学校的位置，一个街道或一个社区的范围。GBase 8t 的空间数据类型也符合 OpenGIS SQL 1.1 版本规范和 ISO/IEC13249-3 SQL/MM 第三部分有关空间信息规范。

GBase 8t 基于 ESRI SDE10.2 shape 和 PE 库提供成熟的 GIS 解决方案。

GBase 8t 支持以下类型的坐标系：地理坐标系和投影坐标系。

GBase 8t 支持不同坐标系之间的相互转换。

1.1 地理坐标系

地理坐标系是使用的经纬度来确定地球上的位置。地球上的任何位置可以通过经纬度坐标点来确定。地理坐标系适合全球数据集的应用，如卫星图像库。

例如，图 1-1 显示了一个地理坐标系中东经 80 度北纬 55 度的位置。

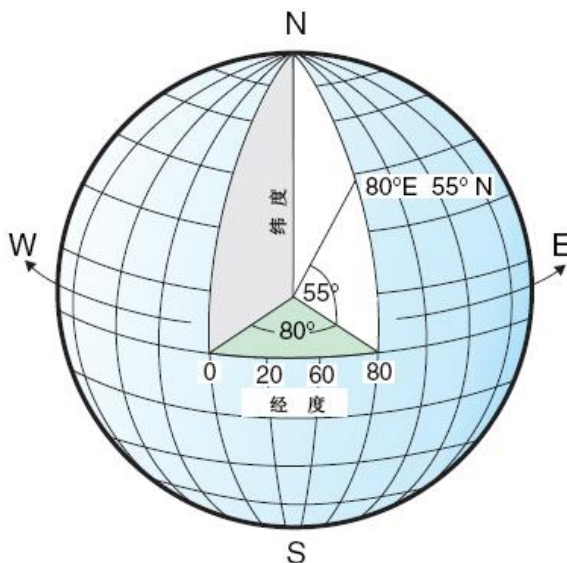


图 1-1 地理坐标系示例

1.2 投影坐标系

投影坐标系统使用二维坐标表示地球上的具体位置。

地理坐标系统通过数学公式计算得到的二维平面投影坐标系。该转换被称为地图投影。

投影坐标系基于椭圆体的地理坐标系，但它使用线性的度量单位，所以计算距离和面积等会变得更为简单，投影坐标系适合区域数据集的应用程序。

2 空间数据解决方案

2.1 应用程序连接空间数据库

图 2-1 展示了应用程序如何访问 GBase 8t 空间数据库。

- Java程序可通过GBase 8t JDBC驱动程序连接空间数据库。
- C应用程序可通过GBase 8t ODBC连接空间数据库。
- ESQL/ C应用程序和DB-Access可直接连接空间数据库。
- ESRI应用通过GBase 8t ODBC驱动程序连接空间数据库。

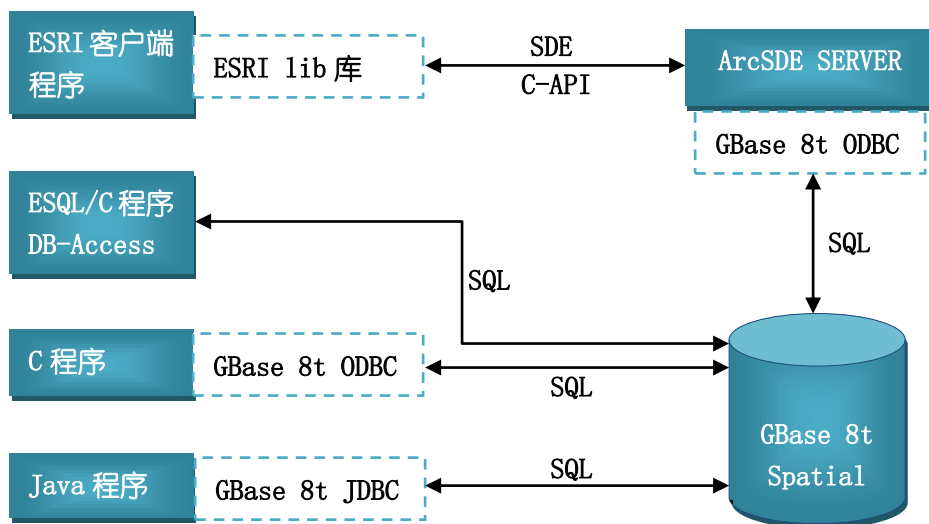


图 2-1 应用程序如何访问空间数据库

2.2 空间数据复制

您可以通过以下环境来复制空间数据库：

- ER (Enterprise Replication) 企业复制
- 高可用集群 (SDS、HDR、RSS)

要复制空间数据库，请注意以下几点：

- 确保spatial_references表实时同步
- 空间数据列必须为非空

2.3 Web 要素服务

GBase 8t 的 web 要素服务 (WFS) 提供对空间数据的 web 交易服务。

WFS 支持以下操作：

- 创建要素
- 删除要素
- 更新要素
- 查询要素

WFS 包括 CGI 客户端和 GBase 8t 为 Web 程序提供的服务函数。这些地理要素以基于 XML 的地理标记语言 (GML) 进行编码。您可以使用 WFS 服务来管理地理数据。WFS 已被写入到开放地理空间联盟 (OGC) 标准 04-094 文件 (网络要素服务实现规范 1.1.0 版)。

3 空间数据库

3.1 spatial_references 表

spatial_references 记录坐标系相关信息，每个空间坐标 ID (SRID) 描述一个特定的空间参考系统。数据库中使用的所有空间坐标系统必须在 spatial_references 表记录。在空间列的所有数据必须使用同一空间坐标系。

spatial_references 表结构说明：

列名	数据类型	示例值	说明
srid	INTEGER NOT NULL	12345	参照系统唯一 id
description	VARCHAR(64)	WGS 1984	该参照系统描述

auth_name	VARCHAR(255)	EPSG	坐标系的标准名称
auth_srid	INTEGER	4326	空间坐标系的id
false_x	FLOAT NOT NULL	-180	x轴参考值
false_y	FLOAT NOT NULL	-90	y轴参考值
xyunits	FLOAT NOT NULL	1000000	缩放比例
false_z	FLOAT NOT NULL	-1000	z轴参考值
zunits	FLOAT NOT NULL	1000	缩放比例
false_m	FLOAT NOT NULL	-1000	M轴参考值
munits	FLOAT NOT NULL	1000	缩放比例
srttext	CHAR(2048)	GEOGCS["GCS_WGS_1984", DATUM["D_WGS_1984", SPHEROID["WGS_1984", 6378137, 298.257223563]], PRIMEM["Greenwich",0], UNIT["Degree", 0.0174532925199433]]	参照系详细描述

表 3-1 spatial_references 表结构说明

3.2 st_units_of_measure 表

度量单位界定米等单位之间的换算系数，st_units_of_measure 表记录度量单位的相关信息。

使用以下函数计算距离或面积时需要使用度量单位：

- ST_Area()
- ST_Buffer()
- ST_Distance()
- ST_Length()
- ST_Perimeter()

使用以上函数时，需要使用表 st_units_of_measure 的 unit_name 列中的度量单位来进行计算。要手动插入数据到 st_units_of_measure，需包含以下列的相关值：

- unit_name: 度量单位名称
- unit_type: 度量单位的类型，可以是linear或angular
- conversion_factor: 度量单位与1米的转换系数
- description: 度量单位的描述

3.3 geometry_columns 表

如果使用 esri 客户端软件，创建一个空间表或往一个表中增加空间列，需要在 geometry_columns 表中插入一行，用于描述空间列使用的坐标系的相关信息。如果使用 shp2sde 或 loadshp 工具往数据库中导入 shape 文件，系统会自动更新 geometry_columns 表，如果要手动插入行到 geometry_columns 表，需要包括以下列的相关值：

- f_table_catalog: 数据库名
- f_table_schema: 数据库模式名
- f_table_name: 表名
- f_geometry_column: 空间数据列名
- geometry_type: 空间数据列类型
- srid: 空间数据列使用的坐标系

表 3-2 为 geometry_type 列中 id 值与空间数据类型的对应关系。

Id	空间数据类型
0	ST_Geometry
1	ST_Point
2	ST_Curve
3	ST_LineString
4	ST_Surface
5	ST_Polygon
6	ST_GeomCollection
7	ST_MultiPoint
8	ST_MultiCurve
9	ST_MultiLineString
10	ST_MultiSurface
11	ST_MultiPolygon

表 3-2 geometry_type id 列说明

3.4 空间数据表

包含一个或多个空间数据列的表被称为空间表。

例如：

```
CREATE TABLE hazardous_sites (objectid integer NOT NULL,
site_id integer,
```

```
name varchar(40),  
location ST_Point)  
PUT location in mysbpace;
```

上表包含一个空间列 `location`，数据类型为点，空间数据存储于智能大对象空间 `mysbpace`。如果未指定空间字段的存储位置，当空间字段长度大于 930Byte 时，自动存储于默认智能大对象空间。

4 空间数据类型

空间数据类型分为两类：基础数据类型和同质集合数据类型。

基础数据类型包括：

- `ST_Point`
- `ST_LineString`
- `ST_Polygon`

同质集合数据类型为同类基础数据类型的集合，包括：

- `ST_MultiPoint`
- `ST_MultiLineString`
- `ST_MultiPolygon`

4.1 数据类型属性

`ST_Geometry` 的数据类型是一个抽象的非实例化的超类，它的子类提供实例化的数据类型。图 4-1 显示了 GBase 8t 的类层次结构空间数据类型。

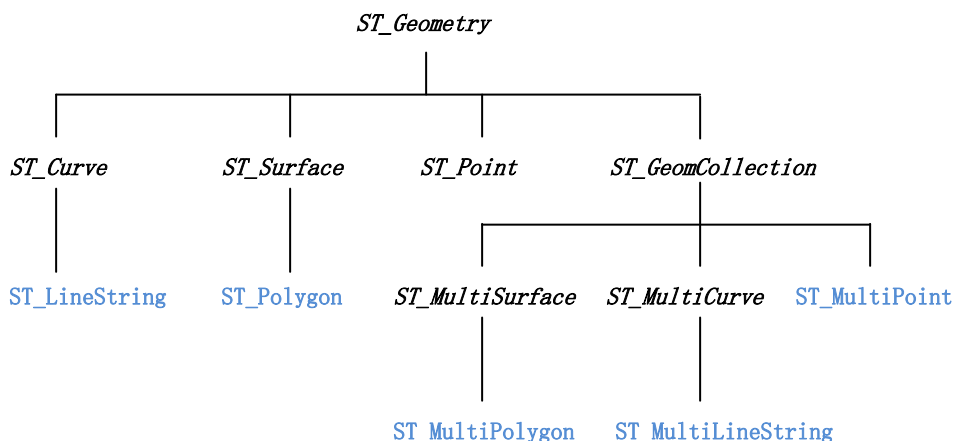


图 4-1 空间数据类型继承关系，黑色斜体为抽象类

可以定义空间数据列类型为 `ST_Geometry`，并可以往该类型数据列插入所有类型的空间数据，但是，定义在子类上的函数可能无法用于该数据类型。

4.2 ST_Point

`st_point` 为坐标系中一个点，用于定义某一个地标，如学校、医院等。

`st_point` 数据类型包含以下属性：

- x坐标
- y坐标
- 可能包含z坐标及m坐标

`st_point` 数据类型有以下函数：

- `st_x()` 返回x坐标
- `st_y()` 返回y坐标
- `se_z()` 返回z坐标
- `se_m()` 返回m坐标

4.3 ST_LineString

`ST_LineString` 数据类型是多个点定义的线性路径。`ST_LineString` 类型一般用来定义的线性特征，如道路，河流和电力线等。

`ST_LineString` 数据类型具有以下属性：

如果线不相交，称为简单线，线的两个端点重叠，称为封闭。

与超类 `ST_Geometry` 相比，`ST_LineString` 具有长度属性。

以下示例描述了 `ST_LineString` 数据类型的特性：

- (1) 简单且不封闭。
- (2) 非简单且不封闭。
- (3) 为简单封闭，形成环。
- (4) 为封闭非简单，不能称为环。

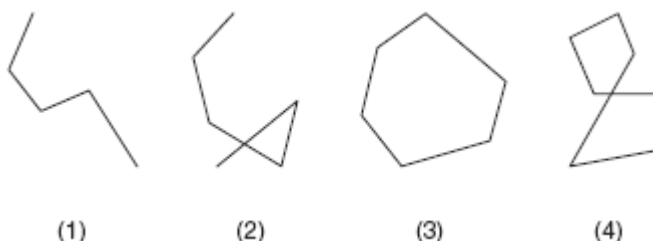


图 4-2 线

ST_LineString 数据类型有以下函数：

- ST_StartPoint() 返回线的起点
- ST_EndPoint() 返回线的终点
- ST_PointN() 返回第n个点
- ST_Length() 返回线的长度
- ST_NumPoints() 返回线点的个数
- ST_IsRing() 是否为环形
- ST_IsClosed() 是否为封闭
- ST_Polygon() 如果线条为环形，返回一个面

4.4 ST_Polygon

ST_Polygon 数据类型是一系列点构成的外环和 0 个或多个内部环组成的面。ST_Polygon 通常描述具有空间的土地，水域，以及其他区域等。

ST_Polygon 数据类型具有以下属性：

外环和内环的边界定义一个面，外环和内环的边界之间的区域为面的内部，面中的环可能相交于某个点，与超类 ST_Geometry 相比，ST_Polygon 有面积的属性。

以下示例描述了 ST_Polygon 数据类型的特性：

- (1) 为由一个外部环组成的面
- (2) 为由一个外部环和两个内部环组成的面，内部环中的区域为面的外部
- (3) 为由一个外部环和相交内部环组成的面

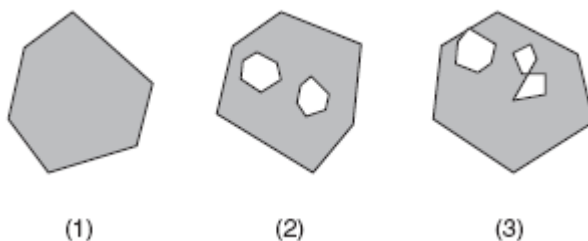


图 4-3 面

ST_Polygon 数据类型的相关函数：

- ST_Area() 返回区域面积
- ST_ExteriorRing() 把外边界返回为环线
- ST_NumInteriorRing() 返回内部环个数
- ST_InteriorRingN() 返回第n个内部环为环线
- ST_Centroid() 返回中心点
- ST_PointOnSurface() 返回面中的一个点
- ST_Perimeter() 返回区域的周长

4.5 ST_MultiPoint

ST_MultiPoint 是一组 ST_Point 的集合，亦称为点簇。

ST_MultiPoint 数据类型具有以下属性：

如果点簇中没有重叠的点，则该点簇为简单点簇。

4.6 ST_MultiLineString

ST_MultiLineString 是一组 ST_LineString 的集合，称为多线串，可用于描述河流、路网等。

ST_MultiLineString 数据类型具有以下属性：

如果集合中的线只在终点相交，称为简单多线串，否则称为非简单多线串。

多线串的境界为不相交的所有线条的终点。

如果集合中所有的线串都是封闭的，那么多线串是封闭的。

与超类 ST_Geometry 相比，ST_MultiLineString 具有长度属性。

以下示例描述了 ST_MultiLineString 数据类型的特性：

- (1) 是简单多线串，它的境界为两条线的四个终点
- (2) 是简单多线串，境界为两个未相交的终点

- (3) 为非简单多线串，边界为三个未相交的终点
- (4) 为简单非封闭多线串，因为它的单条线并未封闭
- (5) 为简单封闭多线串

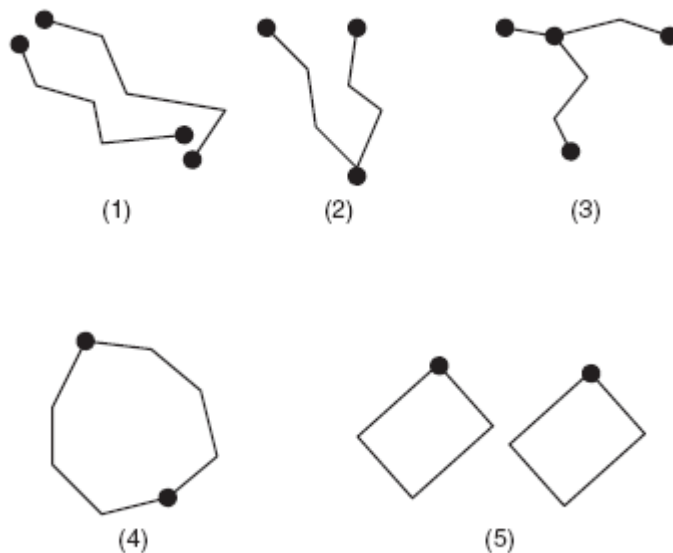


图 4-4 多线串

ST_Polygon 数据类型的相关函数：

- ST_Length() 返回多线串长度
- ST_IsClosed() 判断多线串是否封闭

4.7 ST_MultiPolygon

ST_MultiPolygon 为多个面的集合，用于描述不连续的湖泊、岛屿等信息。

ST_MultiPolygon 数据类型具有以下属性：

ST_MultiPolygon 数据的边界为所有面的边界的总和。

与超类 ST_Geometry 相比，ST_MultiPolygon 有面积的属性。

以下示例描述了 ST_MultiPolygon 数据类型的特性：

- (1) 是一个由两个面组成的区域，边界由两个外环和三个内环构成。
- (2) 是一个由两个面组成的区域，边界由两个外环和两个内环构成。

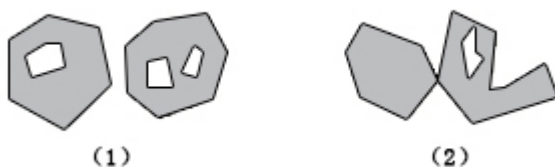


图 4-5 多面

ST_MultiPolygon 数据类型的相关函数：

- ST_Area() 返回区域面积
- ST_Centroid() 返回中心点
- ST_PointOnSurface() 返回区域中的一个点

5 数据转换

5.1 Well-known text (WKT)

GBase 8t 支持从 OGC well-known text (WKT)构造空间数据，WKT 是可以和空间数据进行格式转换的 ASCII 字符格式。

可以使用以下函数构造空间数据或将空间数据转换为 ASCII 字符：

```
ST_GeomFromText()  
ST_PointFromText()  
ST_LineFromText()  
ST_PolyFromText()  
ST_MPointFromText()  
ST_MLineFromText()  
ST_MPolyFromText()  
ST_AsText()
```

5.2 Well-known binary (WKB)

GBase 8t 支持通过 OGC well-known binary (WKB)构造空间数据，WKB 为二进制数据流，它允许应用程序和数据库以二进制形式交互数据。

可以使用以下函数构造空间数据或将空间数据转换为二进制数据流：

```
ST_GeomFromWKB()  
ST_PointFromWKB()  
ST_LineFromWKB()  
ST_PolyFromWKB()  
ST_MPointFromWKB()  
ST_MLineFromWKB()  
ST_MPolyFromWKB()  
ST_AsBinary()
```

5.3 ESRI shape (SHP)

GBase 8t 支持通过 ESRI shape 构造空间数据。

可以使用以下函数构造空间数据或将空间数据转换为 shape：

```
SE_GeomFromShape()  
SE_PointFromShape()  
SE_LineFromShape()  
SE_PolyFromShape()  
SE_MPointFromShape()  
SE_MLineFromShape()  
SE_MPolyFromShape()  
SE_AsShape()
```

5.4 Geography Markup Language (GML)

GBase 8t 支持通过 Geography Markup Language (GML) 构造空间数据。

可以使用以下函数构造空间数据或将空间数据转换为 GML：

```
ST_GeomFromGML()  
ST_PointFromGML()  
ST_LineFromGML()  
ST_PolyFromShape()  
ST_MPointFromGML()  
ST_MLineFromGML()  
ST_MPolyFromGML()  
SE_AsGML()  
ST_AsGML()
```

ST_EnvelopeAsGML()

5.5 Keyhole Markup Language (KML)

GBase 8t 支持通过 Keyhole Markup Language (KML) 构造空间数据。

可以使用以下函数构造空间数据或将空间数据转换为 KML：

SE_EnvelopeFromKML()

ST_GeomFromKML()

ST_LineFromKML()

ST_MLineFromKML()

ST_MPointFromKML()

ST_MPolyFromKML()

ST_PointFromKML()

ST_PolyFromKML()

SE_AsKML()

ST_AsKML()

ST_EnvelopeAsKML()

6 R 树索引

R 树是一个范围树，用于多维数据（空间数据）的查询。

R 树设计用于包含以下数据类型的列：

- 多维数据（二维或三维的空间数据、数组）
- 范围数据，如一个国家在地图上的南北跨度

注意：R 树只能创建一个数据列上

6.1 R 树索引结构

R 树的结构与 B 树的结构类似，R 树有以下特点：

- 基于高度平衡的树结构
- 基于最小矩形边界的方法，用最小矩形包含相关数据
- 叶子节点代表一个单独的区域，包含数据
- 根节点及分支节点不包含数据，分支节点包含最近的叶子节点

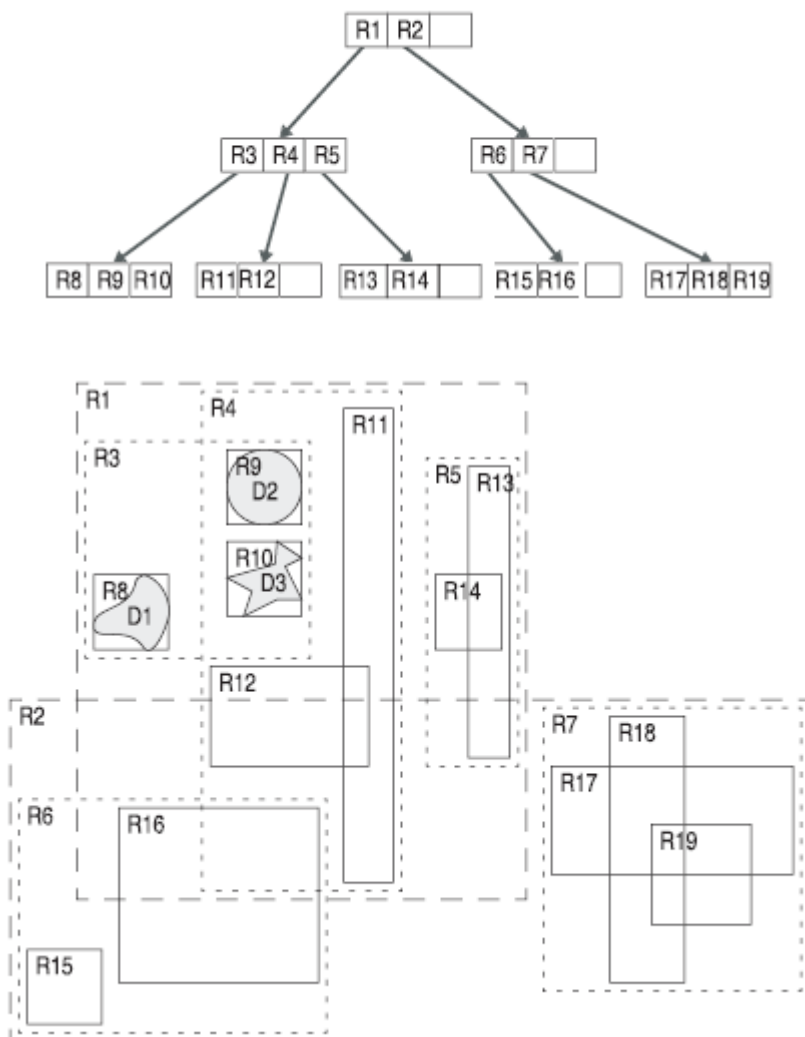


图 6-1 R 树索引结构

如上如所示，数据矩阵可包含一个数据对象或一个或多个数据矩阵，如矩阵 R8，该节点为 R 树的叶子节点，包含数据 D1，矩阵 R3，R 树的枝节点，包含数据矩阵 R8，R9 和 R10。

6.2 创建和使用 R 树索引

使用以下工具导入 shape 数据到 GBase 8t，系统会自动创建 R 树索引：

- loadshp
- shp2sde
- ESRI 桌面软件

要手动创建 R 树索引, 参考以下语法:

```
CREATE INDEX location_idx
ON hazardous_sites (location ST_Geometry_ops) USING RTREE;
ST_Geometry_ops 操作类型包含以下函数:
ST_Contains()
ST_Crosses()
ST_Equals()
SE_EnvelopesIntersect()
ST_Intersects()
SE_Nearest()
SE_NearestBbox()
ST_Overlaps()
ST_Touches()
ST_Within()
```

注意: 不能重命名包含有 rtree 索引的数据库。

7 并行查询

空间数据支持 PDQ 并行计算, 在空间数据使用并行查询能明显地提高性能。

除以下函数外, 所有空间数据函数可使用并行计算:

```
SE_BoundingBox()
SE_CreateSRID()
SE_MetadataInit()
SE_Trace()
```

数据库服务器创建并维护各种数据库对象管理并行空间查询的特征。不要修改或删除以下信息:

```
表 SE_MetadataTable
SE_Metadata 不透明类型
智能大对象空间的元数据
metadata lohandle 文件
metadata 缓存
表 spatial_references 上的触发器
```

如以下信息被破坏, 可使用 SE_MetadataInit() 函数重建以上信息。

8 估算存储空间

空间表的空间大小等于行大小乘以行数加上页头的大小。

要估算空间表大小，需计算以下内容：

- (1) 估算表的行数
- (2) 估算行的长度（空间列和非空间列及页头大小）
- (3) 表的行数和行长乘积即可估算表占用空间大小

估算空间列大小

空间列的大小取决于要素的点数量、坐标系因子、注记大小。

通过以下公式估算空间列大小：

空间列大小=（要素平均点数×坐标系因子）+ 注记大小

坐标系类型	坐标系因子
XY	4.8
XYZ	7.2
XY and measures	7.2
XYZ and measures	9.6

表 8-1 坐标因子

如果图层包含注记，把注记大小以 300Byte 计算。

估算非空间列大小

根据其他非空间数据列的实际字段长度估算空间大小。

估算空间页头信息大小

空间页头信息大小取决于空间表行数，大于 10,000 行的表，每行约 200Byte，1000 到 10,000 行的表，每行约 300Byte，小于 1000 行的表，每行约 400Byte。

估算智能大对象空间

空间数据列大小小于等于 930Byte 时，数据存储于标准数据库空间，当空间数据列大于 930Byte 时，数据列存储指向智能大对象空间的指针，实际数据存储于智能大对象空间。

要估算智能大对象空间大小，根据以下公式计算：

智能大对象比 = (空间数据列大小/1920)

智能大对象空间 = ((智能大对象比) × 数据行数)

估算空间索引大小

根据空间列和页头大小估算索引空间大小。

根据以下步骤估算空间索引大小：

- (1) 将空间列大小和页头大小相加
- (2) 将第一步的计算结果乘以数据行数
- (3) 将第二步的计算结果乘以 1.03

9 空间数据加载与迁移

GBase 8t 提供了以下工具用于管理 ESRI 的 shape 格式文件：

- Infoshp 用于检查shape文件的信息
- Loadshp 用于加载shape文件到数据库
- Unloadshp 用于卸载数据库数据到shape文件

GBase 8t 支持通过 ESRI 客户端桌面软件将 shape 导入到数据库，或将数据库空间数据导出为 shape。

要迁移空间数据，可使用 load/unload, dbexport/dbimport 等迁移工具。

10 空间函数

函数	说明
ST_Area()	返回面或复杂多面的面积
ST_AsBinary()	将空间数据以二进制格式返回
SE_AsGML()	将空间数据以 GML 格式返回
ST_AsGML()	将符合 GML2 或 GML3 标准的空间数据以 GML 格式返回
SE_AsKML()	将空间数据以 KML 格式返回

函数	说明
ST_AsKML()	将空间数据以 KML 格式返回（可设置 KML 属性）
SE_AsShape()	将空间数据以 shape 格式返回
ST_AsText()	将空间数据以文本格式返回
ST_Boundary()	合并空间对象边界
SE_BoundingBox()	将空间列的最大范围以面形式返回
ST_Buffer()	缓冲区计算
ST_Centroid()	返回面的中心点
ST_Contains()	返回第一个空间数据是否包含第二个空间数据的布尔值
ST_ConvexHull()	返回至少包含三个顶点的凸形
ST_CoordDim()	返回空间数据坐标系的维度
SE_CreateSRID()	创建一个坐标系
SE_CreateSrttext()	返回坐标系的详细说明
SE_CreateSrttextCount()	返回同一坐标系的个数
SE_CreateSrttextList()	返回使用某一 id 的坐标系的列表
ST_Crosses()	返回空间数据是否相交
ST_Difference()	返回源空间数据与对比空间数据不一致的地方
ST_Dimension()	返回空间数据维度
ST_Disjoint()	判断两个空间数据是否存在包含关系
SE_Dissolve()	合并同类空间元素为集合类型
ST_Distance()	返回两个空间数据的最短距离
ST_DistanceToPoint()	返回线的起点到指定点的距离
ST_EndPoint()	返回线的终点
ST_Envelope()	返回空间数据的边界
ST_EnvelopeAsGML()	以 GML3 格式返回空间数据的边界
SE_EnvelopeAsKML()	以 KML 格式返回空间数据的边界
ST_EnvelopeFromGML()	根据 GML2 或 GML3 格式数据创建空间数据
SE_EnvelopeFromKML()	根据 KML 格式数据创建空间数据
SE_EnvelopesIntersect()	返回两个空间数据是否相交
ST_Equals()	判断两个空间数据是否相等
ST_ExteriorRing()	将面的轮廓以线格式返回
SE_Generalize()	减少空间数据的点
ST_GeometryN()	返回空间对象在数据集中的位置
ST_GeometryType()	返回空间数据类型
ST_GeomFromGML()	从 GML 格式数据构造空间数据

函数	说明
ST_GeomFromKML()	从 KML 格式数据构造空间数据
SE_GeomFromShape()	从 shape 格式数据构造空间数据
ST_GeomFromText()	从文本数据构造空间数据
ST_GeomFromWKB()	从 WKB 数据构造空间数据
SE_InRowSize()	返回空间数据在标准空间中的行长
ST_InteriorRingN()	返回面的指定块为线环
ST_Intersection()	返回空间数据交集
ST_Intersects()	返回空间数据是否存在交集
ST_Is3D()	返回空间数据是否为 3 维数据
ST_IsClosed()	返回线是否封闭
ST_IsEmpty()	返回空间数据是否为空
ST_IsMeasured()	返回空间数据是否有度量单位
ST_IsRing()	返回线是否为环
ST_IsSimple()	返回空间数据是否为简单数据
ST_IsValid()	返回空间数据是否正确有效
ST_Length()	返回线或线串的长度
ST_LineFromGML()	从 GML 格式数据构造空间数据
ST_LineFromKML()	从 KML 格式数据构造空间数据
SE_LineFromShape()	从 shape 格式数据构造空间数据
ST_LineFromText()	从文本数据构造空间数据
ST_LineFromWKB()	从 WKB 数据构造空间数据
ST_LocateAlong()	将线返回为点簇
ST_LocateBetween()	返回不同点之间的空间数据
ST_M()	返回点的测量单位
ST_MaxM()/ST_MinM()	返回空间数据的最大/最小度量单位
ST_MaxX()/ST_MinX()	返回空间数据的最大/最小 X 坐标
ST_MaxY()/ST_MinY()	返回空间数据的最大/最小 Y 坐标
ST_MaxZ()/ST_MinZ()	返回空间数据的最大/最小 Z 坐标
SE_MetadataInit()	重新初始化智能大对象元信息
SE_Midpoint()	返回线的中点
ST_MLineFromGML()	从 GML 格式数据构造空间数据
ST_MLineFromKML()	从 KML 格式数据构造空间数据
SE_MLineFromShape()	从 shape 格式数据构造空间数据
ST_MLineFromText()	从文本数据构造空间数据
ST_MLineFromWKB()	从 WKB 数据构造空间数据

函数	说明
ST_MPointFromGML()	从 GML 格式数据构造空间数据
ST_MPointFromKML()	从 KML 格式数据构造空间数据
SE_MPointFromShape()	从 shape 格式数据构造空间数据
ST_MPointFromText()	从文本数据构造空间数据
ST_MPointFromWKB()	从 WKB 数据构造空间数据
ST_MPolyFromGML()	从 GML 格式数据构造空间数据
ST_MPolyFromKML()	从 KML 格式数据构造空间数据
SE_MPolyFromShape()	从 shape 格式数据构造空间数据
ST_MPolyFromText()	从文本数据构造空间数据
ST_MPolyFromWKB()	从 WKB 数据构造空间数据
SE_Nearest()	返回最近对象
SE_NearestBbox()	返回最近对象
ST_NumGeometries()	返回同类数据集合的元素个数
ST_NumInteriorRing()	返回面的内部面个数
ST_NumPoints()	返回空间数据点的个数
SE_OutOfRowSize()	返回存储在智能大对象空间的行长
ST_Overlaps()	返回两个空间数据是否相交
SE_ParamGet()	返回相关参数
SE_ParamSet()	设置相关参数
ST_Perimeter()	返回面的周长
SE_PerpendicularPoint()	返回点在线上的垂直投影
ST_Point()	构造一个点
ST_PointAtDistance()	返回距离起始点指定距离的点
ST_PointFromGML()	从 GML 格式数据构造空间数据
ST_PointFromKML()	从 KML 格式数据构造空间数据
SE_PointFromShape()	从 shape 格式数据构造空间数据
ST_PointFromText()	从文本数据构造空间数据
ST_PointFromWKB()	从 WKB 数据构造空间数据
ST_PointN()	返回线中的第 N 个顶点
ST_PointOnSurface()	返回面表面的点
ST_PolyFromGML()	从 GML 格式数据构造空间数据
ST_PolyFromKML()	从 KML 格式数据构造空间数据
SE_PolyFromShape()	从 shape 格式数据构造空间数据
ST_PolyFromText()	从文本数据构造空间数据
ST_PolyFromWKB()	从 WKB 数据构造空间数据

函数	说明
ST_Polygon()	将线环转化为面
ST_Relate()	比较两个空间数据是否符合对比规则
SE_Release()	返回空间模块版本号
SE_ShapeToSQL()	将 shape 文件转换为 sql
SE_SpatialKey()	返回空间数据的排序键
ST_SRID()	返回空间数据的坐标系 id
SE_SRID_Authority()	返回空间数据的坐标系 id 对应的机构
ST_StartPoint()	返回线的起点
ST_SymDifference()	返回空间数据的补集
SE_TotalSize()	返回空间数据的总行长
ST_Touches()	返回两个空间数据是否接触
SE_Trace()	跟踪空间数据函数
ST_Transform()	转换空间坐标系
ST_Union()	合并两个空间数据
SE_VertexAppend()	在线的终点追加一个顶点
SE_VertexDelete()	删除一个顶点
SE_VertexUpdate()	更新顶点
ST_Within()	判断第一个数据是否包含于第二个
ST_WKBToSQL()	将 WKB 转换为 SQL
ST_WKTToSQL()	将 WKT 转换为 SQL
ST_X()	返回点的 X 坐标
ST_Y()	返回点的 Y 坐标
ST_Z()	返回点的 Z 坐标

表 10-1 空间函数说明

GBASE

南大通用数据技术股份有限公司
General Data Technology Co., Ltd.



微博二维码



微信二维码

