

图形数据与关系数据库的结合及其应用

刘仁义¹, 刘 南¹, 苏国中²

(1. 浙江大学 西溪校区地科系, 浙江 杭州 310028; 2. 海南省基础地理信息中心, 海南 海口 570203)

Manipulation of Spatial Graphical Data in RDBMS and
Its Implementation in Application of GIS

LIU Ren-yi¹, LIU Nan¹, SU Guo-zhong²

(1. Xixi Campus, Zhejiang University, Hangzhou 310028, China; 2. Hainan Provincial Geomatics Information Center, Haikou 570203, China)

Abstract It is currently one of the research focuses in the fields of database and GIS for Spatial graphical data to storage in RDBMS. A data model of spatial graphical data storage in RDBMS is presented in this paper. The paper analyses the data organization and operation, the expression of spatial graphical features and system structure in the data model. According to the given data structure model, we realize the operation of storage, topologic arithmetic and topologic analysis for spatial graphical data in RDBMS based on SQL Server. Finally, an application example in land registration system is given by using the data model to implement compact spatial database engine based on SQL Server 7.0.

Key words spatial database model; data organization and operation; system structure

摘 要: 将空间图形数据存入关系型数据库是当前数据库和地理信息系统领域研究的热点和前沿。本文提出了一种将空间图形数据存入关系型数据库的数据模型, 分析了数据模型的数据组织操作、空间图形要素表达以及数据模型体系结构。根据该结构模型, 实现了在 SQL Server 关系型数据库中对空间图形数据进行存储、拓扑运算和拓扑分析的操作。最后, 给出了一个采用该结构模型实现的基于 SQL Server 数据库平台的精简型空间数据库引擎 Spatial Database Guide 软件产品, 及其在土地产权产籍信息系统领域的应用实例。

关键词: 空间数据库模型; 数据组织及操作; 体系结构

1 引 言

关系型数据库是当前商业信息系统运行中的主流数据库系统, 由于发展历史长, 不仅功能强大, 性能稳定, 而且具有完备的工业标准。尤其是像 ORACLE, SQL Server, SYBASE, Informix 等

这样一些企业级的大型关系数据库采用 Client / Server 结构, 所提供的数据安全性、完整性、多用户共享机制是企业级信息系统中必不可少的核心功能。在地理信息系统 (GIS) 的集成环境中, 数据库被 GIS 系统平台完全封装, 并且在同一进程中对数据库进行操作, 效率较高; 不仅具有图形显

示、编辑处理功能,而且提供了强大的空间分析能力,通过内置的属性数据库,将图形与数据库进行关联,实现属性到图形或图形到属性的双向查询。但是,内嵌数据库缺乏商业数据库所拥有的开放性、一致性、安全性和标准化等一系列重要特性,不适合管理海量、网络级的 GIS 信息系统。商业大型数据库虽然功能已非常完善,但不能存储空间图形以满足 GIS 系统的特殊要求。因此,将 GIS 与商业数据库结合,成为数据库和 GIS 界研究的热点之一^[1]。

将空间图形信息存入关系型数据库后,必须能够满足 GIS 系统对空间图形数据的特殊查询和拓扑运算,即根据属性数据访问图形数据,或由图形数据访问属性数据,对图形数据可以进行空间分析、空间统计、图形合并、图形分割等操作。目前,国际上在此领域进行较深入研究并形成软件产品的只有世界著名 GIS 软件开发商美国环境系统研究所 (ESRI—Environmental Systems Research Institute Inc.) 的 Spatial Database Engine (SDE) for Oracle (/Infomix /Sybase /IBM DB2 /SQL Server),美国 MapInfo 公司的 SpatialWare for Oracle (/Infomix /IBM DB2)和 ORACLE 公司的 Spatial Cartridge 产品,其中,支持 SQL Server 7.0 的空间数据库引擎,仅有美国环境系统研究所于 1999 年 5 月推出的一个版本^[2~5]。国内目前在这方面还处在初步研究阶段,实用化的同类软件产品还未见报道。公开发表的国际及国内有关这方面的研究论文极少。从实用性、普及性以及应用前景等方面的因素考虑,针对微软 SQL Server 数据库平台,我们提出了一种将空间图形数据存入关系型数据库的数据模型;根据此模型,开发了一套精简型空间数据库引擎 Spatial Database Guide 软件产品。该产品已在地理信息系统应用领域中得到应用。

2 数据模型组织及操作

在关系型数据库中,对空间图形数据管理必须采用连续数据结构^[2, 3, 6, 7]。本文提出的模型,是将空间数据类型加到关系数据库中,以实现在关系数据表中对空间数据的操作,但不改变和影响数据库内核,仅在现有的数据库表中加入图形索引项,创建空间数据表,供系统调用及访问与其关联的空间数据。如图 1 所示,空间数据存入关系型数据库中,并实现查询的基本概念。

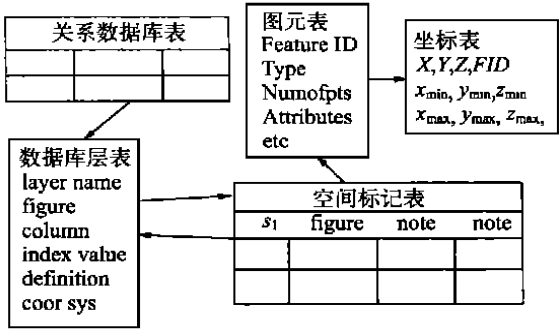


图 1 数据组织及操作

Fig. 1 Data organization and handling

在图 1 中,层表 (Layer Tables)将区域内的地理数据分成不同的专题层,每个专题层的具体数据可以存放在不同的表中,层表记录了区域内空间数据的基本状况,并对其他的表进行管理。通用的商业数据库表,加入图形数据项后,称为空间标记表 (Spatial Stamping Tables)。空间标记表的每一条记录对应一个图元,除了记录若干属性值外,还包含图形的数据字段 (figure)。但实际的空间数据并不放在空间标记表中,而是存放在相关的图元表 (Features Table)中,两者通过 Figure 字段关联。图元表 (Features Table)是记录空间数据的表,它对每一个图元进行编号,记录每一个图元的类型 (点、线、多边形等)、该图元所含点的数量 (对线而言是端点和节点,对多边形而言是顶点)和点序等。但具体的坐标值不放在图元表中,而放在坐标表中,两者通过图元号 (FID)关联。坐标表 (Coordinate Tables)存放每一个图元的点坐标,以及图元最大、最小坐标值,它通过图元号与图元表关联。采用这种方式存储和组织空间数据后,对空间标记表就可以象对待通常的关系表那样进行图形数据的合并、分割和拓扑运算,也可以进行图形到属性或属性到图形的查询^[6~7]。

3 数据模型的空间图形要素表达

采用连续的数据结构,对于所管理的数据对象可以不受范围的限制。可将连续相连的封闭图形构成一个大的封闭图形,然后拼接成整个区域。在数据库中,每一个封闭图形被看作一个完整的独立图形存放,只需一次操作,就能实现对整个图形的提取,具有相同属性的图形存放于同一个图形数据层中。数据模型可将整个层定义为可编辑、可显示、或不可编辑、不可显示等,所有的几何形

状可归纳为点、线、面 3 种基本要素

1. 2 维平面空间点: 定义离散的、无面积或长度的地理要素, 如大比例尺地图上的水井、电线杆以及小比例尺地图上的建筑物等, 其拓扑关系可用单一的 (X, Y) 坐标记录表达

2. 简单线段: 一组有序的 (x, y) 坐标串表示的地理要素, 如街道、河流、等高线等地理要素

3. 面 (多边形): 一组起始点和终止点相同的封闭线段对应的 (x, y) 坐标串或称多边形, 如行政边界、土地利用图地块、地籍宗地等。面可以是简单的多边形或带岛的多边形

2 维坐标值还可以扩展到 3 维 Z 值用来表示 X, Y 点处对应的高度或深度, 图形可以是 2 维的 (X, Y) , 也可以是 3 维的 (X, Y, Z) , 对每一种类型的图形都有一组合法性检查规则, 用以在将该图形存入 RDBMS 之前, 检验其几何正确性。

4 数据模型体系结构

4.1 数据模型实现的主要技术条件

4.1.1 数据模型组织

建立一整套完善的空间数据模型体系, 定义能满足空间查询和分析运算的地理数据类型, 将空间数据合理地存入 SQL Server 关系型数据表中, 并通过双向索引表建立商业数据表与空间坐标表间链接, 解决在一个 Database 内多个属性与空间 table 的关联一致性和可逆性问题

4.1.2 数据空间完整性操作

必须实现标准 SQL 操作下对数据的锁定机制, 解决 Client / Server 结构下对数据的共同操作, 防止用户间破坏数据完整性问题

4.1.3 数据库中实现数据保护

必须确保将空间数据存入数据库中无需进行复杂的应用修改或使用特殊的保护装置, 对用户使用权限的管理能够简单方便地实现。

4.1.4 优化数据库查询

使用基于规则的查询优化技术, 实现对空间数据的有效操作, 提高客户 / 服务器交互处理速度, 达到或超过空间数据库平台的空间查询能力。

4.1.5 建立供开发必需的客户库和商业服务库

建立一套在客户端、商业服务层供开发工具调用的函数库, 即 C-API, 建立空间数据表和属性表的合并关系, 支持空间数据表处理, 即当系统访问空间数据表时, 使用系统提供的 API, 访问属性数据表时使用标准的 SQL 功能^[2,3,8]

4.2 数据模型体系结构

在三级 Client / Server 网络系统中, 数据模型在客户应用端、商业服务端和数据服务器端均能对空间图形数据进行操作, 其体系结构如图 2 所示。

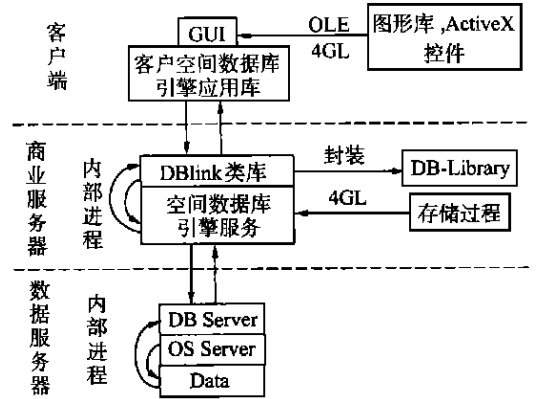


图 2 数据模型的体系结构

Fig. 2 System structure of data model

在客户端可以利用 Visual C++ 调用 ActiveX 控件, 或为某一特别应用直接进行开发, 应用程序调用的客户空间数据库引擎应用库是一组功能强大的类库, 是一个处理客户端应用请求的 API 接口。

商业服务接收应用服务发出的请求, 通过商业服务对空间图形和数据库进行两级封装, 根据确定的商业规则和商业策略, 由空间数据库引擎向数据服务器转发 SQL 指令, 并接收返回结果, 发往客户服务前端。

在服务器端, 通过空间数据库引擎处理模块, 对存入 SQL Server 中的空间图形数据进行访问。服务器在本地执行全部的空间查询和数据提取、存入工作, 将满足搜索条件的数据在服务器缓冲存放并经网络发回到商业服务。缓冲处理收集大块数据, 然后将整个缓冲区中的数据发往商业服务, 而不采用一次只发一条记录的方式。采用这种在服务器端处理并缓冲的方法主要目的是为了提高效率, 降低网络传输压力。

在实际开发应用中, 空间数据库引擎可以采用两种处理方式 (或称为协同处理方式)。处理既可在客户端进行, 也可在服务器端进行。对于一些耗时的计算操作, 可选在客户端处理。

5 数据模型及数据结构实例

本文提出的数据模型能解决空间数据在大型

关系型数据库中的数据存储、更新、查询。空间专题数据的管理是 GIS类软件开发需要解决的主要课题之一。事实上,目前人们普遍认识到的,如:政府确定的行政界限、自然或人为形成的权属范围、土地农业部门确定的土地分类、水利部门管辖的江河流域等等,在 GIS中都是以拓扑多边形或独立多边形的方式进行存储管理的。下面具体说明本文提到的数据模型对具有 shape特性的土地产权产籍数据的管理

5.1 基本数据结构

表 1 属性数据表

Tab. 1 Attribute data table

编号	权利人	土地用途	地价	...
110001	甲	住宅	900	
110002	乙	工厂	1500	
110002	丙	商业	3500	

表 2 空间数据表

Tab. 2 Spatial data table

编号	CX	CY	LBX	LBX	RTX	RTY	Stamp	OBJPRT
110001	485.9	419.5	212.1	161.3	759.8	677.7	1	0xF89001
110002	1063.7	537.6	669.2	160.4	1458.3	914.6	2	0xF89002
110002	684.4	223.3	589.2	160.8	779.7	285.7	3	0xF89003

表 3 空间数据索引表

Tab. 3 Spatial data index table

ID	LBX	LBX	RTX	RTY	numobj	StampNos
1	100	100	200	200	1	01
2	200	100	300	100	2	0102
3	100	200	200	300	1	0102
4	200	200	300	300	9	01030405..

属性数据表和空间数据表就其形式和关系数据库中一般表无任何区别。然而,空间数据表已经包含了空间目标以及一些空间目标的几何特性。其中 (CX, CY)为几何目标的中心点, (LBX, LBX)、(RTX, RTY)为几何目标的外接矩形左下角和右上角坐标, stamp 为土地层目标惟一标志字, OBJPRT为指向空间目标的数据指针。空间数据索引表是在空间数据表更新时动态生存的一张表,其中 (LBX, LBX)、(RTX, RTY)为索引块的地理范围, numobj 为落入该范围的目标数, StampNos为落入该范围的目标标记串。

5.2 数据存储方式

属性数据在不同关系数据库中的存储并无任何特别之处。但是,由于空间数据的数据量大,检索关系复杂,数据的存储方式应依据关系数据库

的本身特性来设计。例如:在 SQL Server数据库中点目标线目标最好存储在 Float 类型的字段中,而线串目标和具有 shape特性的面状目标最好按其数据量的大小分别存储在 VarBinary 和 image 对应类型的数据字段中。

5.3 属性索引

在数据库中实现快速数据查询,与确定数据表的关键字和建立合理的字段索引密切相关。在关系数据库中为配合空间数据的查询检索,属性数据表除满足一般的要求外,还应该与空间数据有惟一的关联字。

5.4 空间索引

按空间位置检索具有某些特性的数据集是 GIS软件的重要特征之一。如何快速定位和检索空间目标主要依赖于合理的空间索引和有效的检索方法。本文提到的数据模型分两个层次建立空间数据库索引。首先,对单个目标的几何特性建立索引满足一般的查询要求,如上表所列使用目标中心点,外接矩形作为基本索引。其次,将整个目标空间分为标准矩形格网块,按四叉树的递归算法建立空间索引,以满足空间目标查询的快速定位和空间分析。

5.5 空间检索

在关系数据库强大的 SQL支持下,对属性数据的检索和查询是较容易实现的。而空间数据的查询必须依托 SQL进行合理设计。例如:查询落在 (100, 100)(500, 500)范围内的空间目标个数可用

Select count(*) from table2 where CX > 100 and CX < 500 and CY > 100 and CY < 500
这种类型的查询只要对目标表进行检索即可;如果给定一个点 (px0, py0)要查找与该点相关的目标应该用以下步骤进行查询

- 1. 从空间索引表中查找第一目标集:

Select StampNos from Table3 where px0 < RTX and px0 > LBX and py0 < RTY and py0 > LBX In To FirstSpatialSets

- 2. 从空间数据表中依据空间目标几何特性查找第二目标集:

Select Stamp from Table2, FirstSpatialSets Where px0 < RTX and px0 > LBX and py0 < RTY and py0 > LBX and Table2 stamp = FirstSpatialSets. stamp In To SecondSpatialSets

3. 从第二目标集依据点和目标的几何关系查找并定位目标:

对于点目标使用查询条件 $\text{Min}(\text{Distance}(\text{px0}, \text{py0}, \text{objx}, \text{objy}))$

对于线目标使用查询条件 $\text{Min}(\text{Perpendicular}(\text{px0}, \text{py0}, \text{Obj}))$

对于面目标使用查询条件 $\text{PointInPology}(\text{px0}, \text{py0}, \text{Obj})$

6 应用成果与结论

根据本文提出的空间数据模型和方法,研制开发了一套基于 SQL Server 7.0 数据库平台的精简型空间数据库引擎 Spatial Database Guide (SDG) 软件产品,实现了在 SQL Server 关系型数据库中对空间图形数据进行存储、拓扑运算和拓扑分析等操作。在商业化的三级客户/服务器结构的土地产权产籍管理系统中得到应用。图 3 为 SDG 在土地产权产籍管理系统运行的实例。系统对地理实体和数据库连接分别进行封装,地籍图形数据通过空间数据库引擎 SDG 存入 SQL Server 数据库中,系统实现了对土地产权产籍管理工作全过程的动态管理。对宗地的变更历史可以进行快速浏览,对于已成为历史状态进入历史库的数据,也可方便地进行图形属性双向互查。系统可以按变更的历史顺序依次撤消,将最后一次认可的历史数据回退到现时库中(时空数据模型是该研究的另一重点,这里不进一步展开)。此外,系统还提供了图形编辑、输出、数据自动备份、系统修复、互联网信息查询、系统维护及安全保护等功能。在土地产权产籍系统中的应用表明,空间数据库引擎 SDG 运行稳定,性能良好,易于开发,适用面广。在本室承担的水利等其他 GIS 应用领域也得到了很好的应用,替代了昂贵的进口同类产品。但 SDG 仅是一个精简型的空间数据库引擎,总体功能还有待于进一步完善,目前正在进行这方面的研究开发和改进工作。



图 3 土地产权产籍管理系统运行实例
Fig. 3 Application example of SDG in land registration management system

参考文献:

- [1] GONG Jian-ya. Some of Theories and Technology about GIS in the Contemporary Era[M]. Wuhan: Publishing House of Wuhan Technical University of Surveying and Mapping, 1999. 44-46. (in Chinese)
- [2] ESRI. Spatial Database Engine[EB/OL]. <http://WWW.ESRI.COM>. 1999-10-22.
- [3] MAPINFO. SpatialWare[EB/OL]. <http://WWW.MAPINFO.COM>. 1999-10-22.
- [4] INTERGRAPH. Software Product[EB/OL]. <http://INTERGRAPH.COM>. 1999-10-22.
- [5] ORACLE. Data Cartridge Operating System Interface[EB/OL]. <http://WWW.ORACLE.COM>. 1999-10-22.
- [6] SA Shi-xuan, WANG Shan. Conspectus of Database System (2nd ed) [M]. Beijing: Higher Education Publishing House. 1992. 19-35. (in Chinese)
- [7] YAN Wei-min, WU Wei-min. Data Structure (2nd ed) [M]. Beijing: Tsinghua University Publishing House. 1992. 19-35. (in Chinese)
- [8] TOM H. GIS Database Directions [A]. Geoinformatics '98 Conference Proceedings [C]. Beijing: China Association for GIS. 1998, 79-88. (in Chinese)