

空间数据的表示与动态分割技术

何心远 邱名卿 (深圳市福田区税务局 518048)

摘要:由于空间数据的非结构化特点,传统的管理方法是采用文件系统,导致数据一致性、完整性及安全性等问题。如何利用数据库技术管理空间数据是同行关注的问题。Oracle7 通过引入 HHCODE 数据类型和动态分割技术成功地解决了空间数据的描述和管理问题,实现了在单一数据库中存取及管理不同类型的数据。本文拟详细分析和讨论该技术。

关键词:空间数据 Oracle7 Regions 数据库 GIS 动态分割。

一、引言

空间数据是指具有空间含义的多维数据,它有广泛的应用领域,如(长度、宽度、高度、质量)四维数据表示一个物体的物理特征,而(经度、纬度、海拔)表示一个地理位置。利用计算机技术管理空间数据早得以应用,典型的例子是 GIS(地理信息系统)。由于空间数据量无法预计及动态增长等非结构化特点,传统管理方式是采用文件系统,如 Genamap,而空间数据的描述信息即属性数据仍采用 RDBMS,如 Oracle,两者之间用关键字或指针完成联系。这种分离管理方式导致以下弱点:

- * 数据完整性和一致性不能保证。
- * GIS 的开放性及其互操作性受限制。
- * 数据共享和并行处理无保证,不能支持 Client/Server 结构。
- * 缺乏数据动态增长能力和数据优化管理。
- * 查询速度慢。

Oracle7 Spatial Data Option(空间数据选件,以后简称“选件”)是运行在 Oracle 内核上管理空间数据的产品,与 RDBMS 并存。它通过引入新的数据类型和动态分割技术,继承了传统 RDBMS 的优点,很好地解决了以上问题,完成了属性数据和空间数据的一体化管理。以下详细分析和讨论“选件”中的管理技术。

二、空间数据的表示与组织

1. HHCODE 数据类型

Oracle7 的空间数据管理技术是在数据库中引入了一个新的数据类型: HHCODE (Helical Hyperspatial Code),它能把具有 n 维空间的矢量数据转化为单个线性

数值,这个数值表示矢量数据在多维空间中的全部含义,它在空间数据表中占一个列。

HHCODE 的编码方法:

(1) 将 n 维对象空间的 n 个坐标轴轮流作为分割轴进行二等分递归操作,直至分割形成的超立方体内仅包含属于单一的空间细胞为止。

(2) 用 2 树来管理上述分割过程。以 0, 1, 2, ... 2 个编号,按固定顺序(如顺时针)对每个超立方体节点进行编号,根节点编号为空,层次为 0。

(3) 某个超立方体的 HHCODE 值就是从树根节点到该超立方体所经过节点的编号顺序拼连(从左到右)。

图 1 是一幅世界地图,现用四层四叉树来分割该二维空间领域(Regions),图 2 是分割后各层子领域的 HHCODE 编码。不难看出,HHCODE 的长度代表领域的分割层数,位数越长,表示的空间领域越小,即越精确。它具有以下特点:

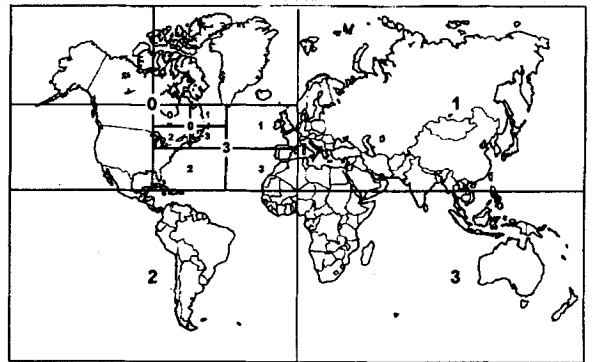


图 1 四层分割二维空间

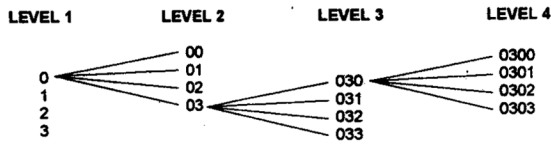


图 2 各层子领域的 HHCODE 编码

* 空间描述性。二维空间中每个 HHCODE 表示一个平面区域;三维空间代表一个立方体;依次类推, n 维空间中表示一个 n 维空间领域(超立方体)。

* 准确性。HHCODE 编码可以根据空间数据的精度要求不断递归分割(即增加 HHCODE 的位数),以逼近数据的精确位置。

* 遵循空间数据表示的工业标准。在几乎所有的 GIS 系统中,都支持 HHCODE 的访问及转换,如 Mapinfo 和 Arcinfo。因此,这些 GIS 系统能直接与 Oracle7 连接,直接将属性数据和空间数据存入统一的 Oracle7 中。

* 表示领域范围的可变性。通过截取 HHCODE 的长度,则可以表示不同范围的空间领域。

2. 基本数据类型的 HHCODE 表示

(1)点(Points)。二维空间中的点可以采用 HHCODE 编码,HHCODE 的长度及分辨层次来定义点的大小。图 3 是二个不同大小的点的 HHCODE 表示。

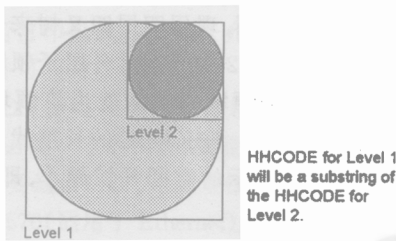


图 3 二维空间中点的 HHCODE 表示

(2)线(Lines)。二维空间的直线可用两端点的 HHCODE 值来表示,见图 4。



图 4 直线的 HHCODE 表示

属于两个或多个网络中的直线,由于直线的方向不同,要用两个 HHCODE。图 5 中,多边形 A 和 B 共同拥有由点 1 和点 2 组成的直线,但是该直线在 A 和 B 中的方向不同,故要由两个 HHCODE 表示。

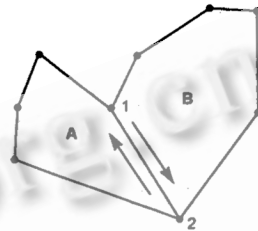


图 5 网络及多边形中直线的方向

(3)领域(Regions)。HHCODE 的空间描述性决定了任何空间领域可用 HHCODE 表示。为了客观描述一个领域,往往要根据所需精度要求由一个或一组 HHCODE 表示。

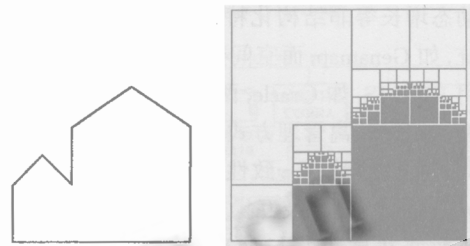


图 6 二维领域的 HHCODE 编码

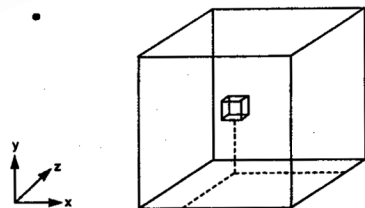


图 7 三维空间的 HHCODE 表示

图 6 所示,左边的多边形领域能用右图所有黑色领域的 HHCODE 值描述。在图 7 中,HHCODE 表示一个六面体。

3. 维数(Dimensions)的定义

空间表中定义 HHCODE 之前,应先定义多维空间系统,包括以下内容:

- * 空间维数的数量和类型。
- * 每维的坐标系统。
- * 每维的上界和下界范围。
- * 每维的比例尺。

如用 HHCODE 表示二维空间的经度和纬度,空间维数的定义见表 1。

表 1 二维空间的维数定义

Dimension Number	Lower Boundary	Upper Boundary	Scale
1 (Latitude)	-90 degrees	+90 degrees	7
2 (Longitude)	-180 degrees	+180 degrees	7

4. 动态分割技术

数据库中数据组织原则是:既要能存储大量数据,又要能快速检查。为此,传统的优化方法是将大量的数据平均分配到不同的表中,并且限制查询表的数量和表中记录个数。至今,数据库设计者只能通过估算数据量来确定表的个数以及如何更有效分配数据在此表中。这种方法存在两个问题:

- * 积累的数据量无法预先确定。
- * 数据量增长可能导致多次数据库重构和数据加载。

“选件”采用了具有独特之处的动态数据分割(partitioning)技术,因而,VLDB 的设计者不再需要预先估算数据量及如何均匀分配数据到各表中以利于优化查询。

动态分割技术是利用 HHCODE 的结构特点将数据分割后存放在多个表中,称为分割表(partitioned table),每个分割区代表多维空间中一个特定的领域。分割可以递归,当分割区中的数据量超过指定最大数据量时,该区又将进一步细分。首次分割表数为 1,称为根表,每个表可细分的表数为 2 个(n 是空间维数),称为孩子表。整个分割过程是在数据加载中自动进行的,数据库设计者仅需指定表大小,系统完成表分割及聚集存放工作。

5. 空间数据表

存放空间数据的表称为空间数据表,分为两类即:分割表和非分割表。其分类基于以下原则:

- * 数据量易于管理的用单个表即非分割表。
- * 数据量太大或不能预测的采用分割表。

(1)非分割表。与传统关系表的特点一样,所不同的是该表中有 HHCODE 域。所有空间数据存放在单个表

中,数据之间是通过 HHCODE 域来维护其间的关系,通过它建立独立索引和聚集关系。

(2)分割表。对数据量大和无法预测量的情况,就必须创建分割表来解决。分割表设有“可分割”标识,HH-CODE 域作为分割键。

数据最初被加载到 2 个分割表中,每个分割表的结构、大小、域数、域的类型及维数的定义与其父亲表完全相同。分割表之间形成 2 树型结构, n 为 HHCODE 编码的空间维数。表数可以动态增加,分割表的数量根据空间数据量、领域的数据密度以及空间维数确定。“选件”数据字典中记录了描述表之间隶属关系以及表中记录之间共同的 HHCODE 子串的所有信息,以便快速查找。

创建分割表时需指定表中能存放的最大记录数,称之为最高水位(High water mark)。数据加载超过最高水位时,就得进一步细分产生新的分割表(孩子表)。

6. 表分割过程

图 8 是三维空间分割过程,第一层中数据加载到 8 个或更少的分割表中,无数据的分割表实际上并没有被创建,但要在“选件”数据字典中加以说明,当这些表所表示的数据空间有数据加载时,它们才被真正创建。根表为空,永不删除。HWM(High Water Mark)表示最高水位,当加载数据超过 HWM 时,该表进一步分割为另外 8 个子表。只有装有数据的表才被创建。最终存在的分割表集仅是装有数据的表,图 8 中的阴影部分就是数据加载完毕后真正创建的分割表集。

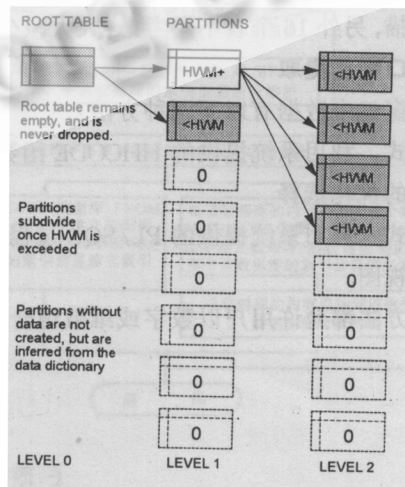


图 8 三维空间的表分割过程

三、空间数据存取及操纵

1. 数据存取

传统 RDBMS 中数据存取要搜索整个数据库或索引,而“选件”仅仅是利用分割表和数据字典查询相关表,检索时间大大缩短,并且减少查询时所需存取空间,与层次空间索引结构不同的是存取语言采用标准及扩展 SQL 语言。“选件”数据字典中为每一个分割表登记一个条目。

数据存取时,应完成以下工作:

* 用户指定感兴趣的领域“窗口”。* 系统检索“选件”数据字典,确认窗口所覆盖的分割表。

* 扫描这些分割表,提取符合查询条件记录集合传递给应用程序。

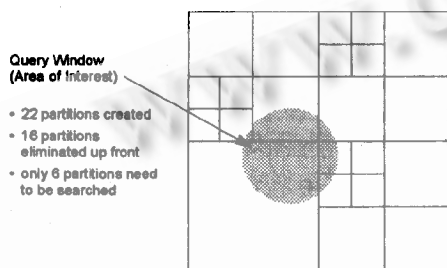


图9 查询区域及对应分割表

图9所示,整个二维空间需22个分割表,圆形区域为所需查询窗口,只覆盖6个分割表,查询时仅对该6个表进行扫描,另外16个表不用考虑,大大减少查询时间。

2. 窗口数据提取

提取窗口中数据有以下两种方法:

* 显式。利用系统提供的 HHCODE 函数提取数据,并有完整的优化选择。

* 隐式。利用系统提供的 PL/SQL 过程来创建窗口数据表或视图。

两种方法都允许用户以数字或维数组合提取。窗口

可有以下三种方式:

* 二维矩形。用于搜索指定范围内的目标。

* 二维圆。寻找最近距离对象时采用。

* 二维多边形。搜索封闭多边形内目标时使用。

3. 数据操纵

“选件”提供了扩展的 SQL 语言完成空间数据的操纵、查询以及分析工作,它能允许用户形式化特定的查询,为 RDBMS 中建立复杂应用提供基本支持。扩展的 SQL 部分包括以下功能:

* 数据定义语言 (DDL): 空间数据表的 Create、Alter、Lock 和 Drop。

* 数据操纵语言 (DML): 空间数据表的 Selete、Delete、Insert、Update 以及分割表维护操纵。

* 数据提取: 从空间数据表中选择数据子集。

* HHCODE 函数集[3]。

在 Oracle7 环境中,以上这些过程和函数可与其他 SQL 命令和函数联合使用。

四、结论

Oracle7 的空间数据管理技术解决了空间数据管理上的两个基本问题:一是属性数据和空间数据的一体化管理;二是动态增长及超大数据量的空间数据的存取。在解决这些问题时引入的 HHCODE 数据类型及动态分割技术,使得空间数据具有自动索引功能,减少了传统 RDBMS 在索引上的开销,确保了在 VLDB 中高速和稳定的查询性能,即:查询数据只需对感兴趣区域的分割表进行扫描,查找时间只依据所查数据集的大小线性增长,而不会考虑数据库的大小。由于在同一数据库中采用统一的 SQL 语言对两类数据进行定义和操纵,还能减少系统的开发成本和后期维护。该技术还遵循开放 GIS 的工业标准。

(来稿时间:1996年9月)