

ORACLE 空间数据库的对象—关系模式初探

——兼议关系数据库和面向对象数据库在 GIS 中的应用

李振华

(中国地质大学信息工程学院, 湖北武汉 430074)

摘要: 传统的关系数据库在 GIS 的应用中主要存在以下问题: 除数据类型简单, 不能很好地表达非结构化数据外, 也难以表示 GIS 中具有复杂结构的数据; 关系模型简单, 不能表达数据之间的层次, 继承、聚合、泛化或特化等在 GIS 广泛使用的关系; 与数据操纵语言如 SQL 和通用程序设计语言, 特别是目前流行的面向对象设计语言之间失配。采用面向对象技术可以较好地解决上述问题。ORACLE 的空间数据库在传统的关系数据库上进行了面向对象的扩展, 即所谓对象—关系模式, 其核心在于增加了名为 SDO_GEOMETRY 的对象数据类型, 用于存储几何实体。ORACLE 的空间数据库的应用示例表明: 以面向对象的形式表示空间数据比之于关系数据库中存放大量无意义的坐标有着很大的优势, 同时, 它在复合对象的表达方面, 在对于空间关系的处理方面都非常有效。

关键词: 面向对象数据库; 空间数据库; 关系数据库; 地理信息系统; ORACLE; 对象—关系模型。

中图分类号: TP311.13

文献标识码: A

文章编号: 1000-2383(2002)03-0333-05

作者简介: 李振华(1969—), 男, 讲师, 1994 年毕业于中国地质大学(武汉), 获硕士学位, 主要从事计算机及地学应用的教学与科研工作。E-mail: zhli@cug.edu.cn

目前的 GIS 的底层数据库大都是关系数据库, 但关系数据库并不能适应目前飞速发展的 GIS 应用需求, 尤其不能适应“数字地球”和“数字国土”、“数字区域”建设中对复杂类型数据和海量数据的管理和处理^[1~4]。因此, 寻求新的数据库模型, 或者对原有数据库进行改造, 已成为当务之急。

1 关系数据库的局限性

(1) 数据类型简单。目前商用数据库系统只支持有限的几种数据类型, 如整数、浮点数、字符串、日期、货币等; 只能表达结构化的数据, 但随着应用的日益发展, 非结构化或半结构化的数据占有越来越重要的地位。在 GIS 实际应用中, 大部分数据都是半结构化甚至非结构化的, 例如属性、文献资料、图纸、合同、传真、E-mail 等都是 GIS 常用的数据。另

外关系数据库对 GIS 中具有复杂结构的数据也难以表达。如几何实体数据, 无论是用基本的原始几何实体组合来描述还是用几何实体边界来描述, 其结构都将是复杂的、层次化的^[5]。总之, 由于关系数据库所固有的结构化的特点, 使其不能灵活地表达现实生活中的各种数据, 为弥补这一点, 在新版本的关系数据库中, 一般只增加一个“二进制”类型(如 blob 或 image 等)来包办用户所需要的其他所有数据类型。这显然不是一个很好的数据表达办法。

(2) 关系模型简单。在关系模型, 实体间的关联无非是一对一、一对多和多对一。然而, 在 GIS 的数据中, 数据之间的关系远非如此简单, 数据之间更多地表示为层次、继承、聚合、泛化或特化等特点。例如, 一个复合对象由多个子对象组成, 如果硬要将这种情况用关系表达, 其结果常常是带有许多冗余数据的不自然的复杂查询, 进一步讲, 重构复杂对象还带来另一个问题, 那就是, 由连接构造的视图一般是不可更新的^[5]。

(3) 阻抗匹配问题。自关系模型产生以来, 阻抗

匹配就一直存在. 这一经典的问题原意是指数据操纵语言如 SQL 和通用程序设计语言之间的失配. 这种不匹配表现在 2 个方面: 一是编程模式不同, 描述性的 SQL 语言与指令式的编程语言如 C 语言不同; 二是类型系统不匹配, 编程语言不能直接表示诸如关系这样的数据库结构, 在其界面就会丢失信息. 进一步说, 由于是 2 个类型系统, 自动的类型检查也成了问题^[5]. 此外, 还须考虑这样一个现实: 面向对象技术已成为主流技术的今天, 面向对象的程序设计语言已成为编写数据库应用的主流语言, 这就严重存在着关系范式和面向对象范式之间的失配问题. 而实现程序设计语言和数据库语言的无缝集成一直是数据工作者所追求的目标.

2 面向对象数据库在 GIS 中的应用

从关系数据库的局限性可以看出, 传统的关系数据库已不能适应 GIS 发展的需要, 必须引入面向对象的技术来解决上述问题. 目前, 面向对象数据库的设计有 2 种形式: 一种是纯面向对象的、与传统关系数据库毫无联系的数据库, 在这方面, 对象数据库管理小组 (ODMG: object data management group) 一直在从事对 C++ 和 SMALLTALK 语言的持久化扩展进行标准化的工作; 另一种是在传统关系库中加入面向对象的特征, 来对其进行扩展, 这种形式目前在实际应用中占主导地位, 可称之为对象—关系数据库.

目前的数据库中, ORACLE 的空间数据库是第 1 个可商用的产品, 下面以其流行的版本 ORACLE8i 为例来进行说明.

2.1 ORACLE 的对象—关系模式

ORACLE 的空间数据库在传统的关系数据库上进行了面向对象的扩展, 即所谓的对象—关系模式, 其核心在于增加了名为 SDO_GEOMETRY 的对象数据类型, 用于存储几何实体. 一个几何实体在表中存为—行, 用列 SDO_GEOMETRY 表示, 同时, 沿用关系数据库中基本的语句 DLL (CREATE, ALTER, DROP) 和 DML (INSERT, UPDATE, DELETE) 来产生和管理空间索引.

SDO_GEOMETRY 这一对象类型是这样定义的:

```
CREATE TYPE SDO_GEOMETRY AS OBJECT (
```

```
SDO_GTYPE NUMBER,  
SDO_SRID NUMBER,  
SDO_POINT SDO_POINT_TYPE,  
SDO_ELEM_INFO MDSYS.SDO_ELEM_INFO_ARRAY,
```

```
SDO_ORDINATES MDSYS.SDO_ORDINATE_ARRAY);
```

其中: SDO_GTYPE 表示几何实体的类型, 以 4 位数 ($d00x$) 形式表示, d 表示维数, 取值为 2, 3 或 4; 后面的位数表示几何实体类型. $d000$ 表示未知, $d001$ 表示点, $d002$ 表示线, $d003$ 表示多边形, $d004$ 表示集合, $d005$ 表示复合点, $d006$ 表示复合线, $d007$ 表示复合多边形. 例如: 2003 表示一个 2 维的多边形. 另外, $d008—d099$ 为预留给值. SDO_SRID 为预留给值, 在空间数据引用系统定义表中, 预备作外关键字使用. SDO_POINT_t 为一个具有属性 x , y 和 z 的对象类型, x , y 和 z 均为数值型. SDO_ELEM_INFO 是一个变长的数值型数组, 用于解释存储于 SDO_ORDINATES 中的坐标. 它以三元组的形式在数组中存放数据, 其形式为 (SDO_STARTING_OFFSET, SDO_ETYPE, SDO_INTERPRETATION), 其中: SDO_STARTING_OFFSET 存放这个元素第一个坐标在 SDO_ORDINATES 中的偏移量. 偏移量的取值从 1 开始, 第 1 个元素的第 1 个坐标将在 SDO_GEOMETRY.SDO_ORDINATES (1) 中, 如果存在第 2 个元素, 它的第 1 个坐标将在 SDO_GEOMETRY.SDO_ORDINATES(n) 中, 这里 n 表示在 SDO_ORDINATE_ARRAY 中定义的位置. SDO_ETYPE 表示元素的类型, 可用 1 位或 4 位数字表示, 若不知道元素是否是内部或外部, 用 1 位数字表示, 否则用 4 位数字表示. 1 位数字取值 1~5, 其中 1~3 表示是简单元素, 4 和 5 表示是复合元素. 4 位数字在 1 位数字取值前加上了“内外标识”, 外部标识取值 1 或 2, 1 表示外部, 0 表示内部, 如 1003 表示外多边形环, 2005 表示内复合多边形环. SDO_INTERPRETATION: 如果 SDO_ETYPE 的值表示的是复合元素 (4 或 5), 那么它的值表示后面有多少个后继的三元组; 如果 SDO_ETYPE 的值表示的是简单元素 (1~3), 那么它的值表示对这个元素的坐标序列进行解释. SDO_ORDINATES 是一个变长的数值型数组 (1048576), 用于存储空间对象的边界坐标, 它通常和 SDO_ELEM_INFO 一起使用.

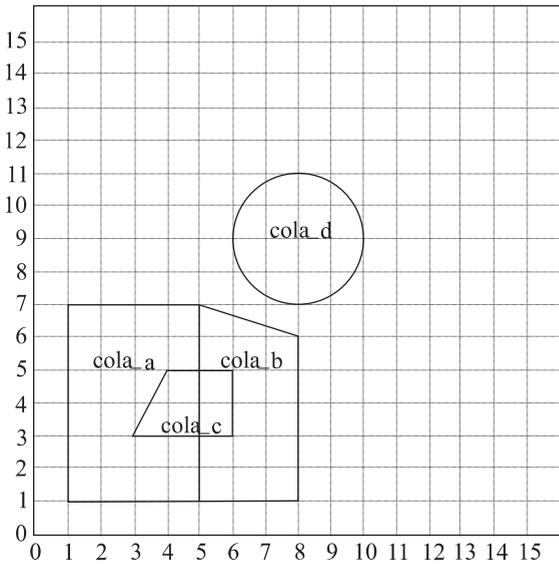


图 1 Four areas of interest for the example

Fig.1 4 个感兴趣的区域

2.2 实例:插入、索引和查询空间数据

图 1 显示了 1 个可乐(colas)厂商感兴趣的 4 个区域,分别标记为 cola_a, cola_b, cola_c, cola_d, 以代表不同的兴趣区域.

(1)建立表 cola_markets.

```
CREATE TABLE cola_markets (
mkt_id NUMBER PRIMARY KEY,
name VARCHAR2 (32),
shape MDSYS. SDO_GEOMETRY);
```

此表存储可乐厂商感兴趣的区域. 其中 SDO_GEOMETRY 是一个对象类型.

(2)将 4 个区域数据插入表中.

①插入区域 a, a 正好是 1 个长方形.

```
INSERT INTO cola_markets VALUES (
1,
'cola_a',
MDSYS. SDO_GEOMETRY (
2003, --表示是 2 维的多边形
NULL,
NULL,
MDSYS. SDO_ELEM_INFO_ARRAY (1,
1003, 3), --1 个长方形(1003 中的首位数字
--1 表示是外部)
MDSYS. SDO_ORDINATE_ARRAY (1,1,
5,7) --只需两个坐标即可定义 1 个长方形
)
```

```
);
②插入区域 b 和 c, 它们是简单的多边形.
INSERT INTO cola_markets VALUES (
2,
'cola_b',
MDSYS. SDO_GEOMETRY (
2003, --表示是 2 维的多边形
NULL,
NULL,
MDSYS. SDO_ELEM_INFO_ARRAY (1,
1003, 1), --1 个多边形
MDSYS. SDO_ORDINATE_ARRAY (5,1,
8,1, 8,6, 5,7, 5,1)
)
);
INSERT INTO cola_markets VALUES (
3,
'cola_c',
MDSYS. SDO_GEOMETRY (
2003,
NULL,
NULL,
MDSYS. SDO_ELEM_INFO_ARRAY (1,
1003, 1),
MDSYS. SDO_ORDINATE_ARRAY (3,3,
6,3, 6,5, 4,5, 3,3)
)
);
③插入区域 d, d 是一个半径为 2 的圆形.
INSERT INTO cola_markets VALUES (
4,
'cola_d',
MDSYS. SDO_GEOMETRY (
2003,
NULL,
NULL,
MDSYS. SDO_ELEM_INFO_ARRAY (1,
1003, 4), --圆
mdsys. sdo_ordinate_array (8,7, 10,9, 8,11)
)
);
(3)更新元数据视图. 建立空间索引前必须更新系统的元数据视图 user_sdo_geom_metadata, 每个
```

图层只需做一次。

```
INSERT INTO USER_SDO_GEOM_METADATA
DATA
VALUES (
'cola_markets',
'shape',
MDSYS.SDO_DIM_ARRAY ( -- 20x20 网
格,容忍度为 0
MDSYS.SDO_DIM_ELEMENT ('x', 0, 20,
0.005),
MDSYS.SDO_DIM_ELEMENT ('y', 0, 20,
0.005)
),
NULL -- SRID 作预留用
);
```

(4) 产生空间索引。

```
CREATE INDEX cola_spatial_idx
On cola_markets (shape)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS ('SDO_LEVEL= 8');
```

(5) 执行某些空间查询。

① 返回 2 个几何实体的拓扑交：

```
SELECT SDO_GEOM.SDO_INTERSEC-
TION (c_a.shape, m.diminfo, c_c.shape, m.
diminfo)
FROM cola_markets c_a, cola_markets c_c,
user_sdo_geom_metadata m
WHERE m.table_name='COLA_MARKETS
' AND m.column_name='SHAPE'
AND c_a.name='cola_a' AND c_c.name='cola_
c';
```

② 查询 2 个几何实体的空间关系：

```
SELECT SDO_GEOM.RELATE (c_b.
shape, m.diminfo, 'anyinteract',
c_d.shape, m.diminfo)
FROM cola_markets c_b, cola_markets c_d,
user_sdo_geom_metadata m
WHERE m.table_name='COLA_MARKETS
' AND m.column_name='SHAPE'
AND c_b.name='cola_b' AND c_d.name='cola
_d';
```

③ 返回所有的区域：

```
SELECT c.name, SDO_GEOM.SDO_ARE-
```

```
A (c.shape, m.diminfo)
```

```
FROM cola_markets c, user_sdo_geom_meta-
data m
```

```
WHERE m.table_name='COLA_MARKETS
' AND m.column_name='SHAPE';
```

④ 返回两个几何实体之间的距离：

```
SELECT SDO_GEOM.SDO_DISTANCE (c_
b.shape, m.diminfo, c_d.shape, m.diminfo)
```

```
FROM cola_markets c_b, cola_markets c_d,
user_sdo_geom_metadata m
```

```
WHERE m.table_name='COLA_MARKETS
' AND m.column_name='SHAPE'
```

```
AND c_b.name='cola_b' AND c_d.name='
cola_d';
```

⑤ 查询某个几何实体是否有效：

```
SELECT c.name, SDO_GEOM.VALI-
DATE_GEOMETRY (c.shape, m.diminfo)
```

```
FROM cola_markets c, user_sdo_geom_meta-
data m
```

```
WHERE m.table_name='COLA_MARKETS
' AND m.column_name='shape'
```

```
AND c.name='cola_c';
```

综上所述,以面向对象的形式表示空间数据比之于关系数据库中存放大量无意义的坐标有很大的优势,同时,它对复合对象的表达,对于空间关系的处理也非常有效。

3 面向对象数据库应解决的一些问题

(1) 从面向对象数据库的发展来看,主要有 2 个方向:一种是以面向对象技术为基础,而引入数据库技术,即所谓“纯”面向对象数据库;另一种是以现有关系数据库为基础,加入面向对象的特征,即所谓对象—关系数据库。(2) 关系模型与面向对象模型毕竟是 2 个不同的数据模型,鉴于目前大部分商用数据库都是关系数据库这一现实,不管是“纯”面向对象数据库,还是对象—关系数据库,暂时都回避不了这 2 种模型的转换或兼容问题。(3) 面向对象的数据模型目前还不像关系数据模型一样有完美的数学基础,其数据库语言还缺乏形式化基础^[5]。(4) 如何将成熟的关系数据库技术,如事务控制、完整性检查、数据复制等应用到面向对象数据库中,也是一个值得研究的方向。

参考文献:

- [1] 李超岭,张克信. 基于 GIS 技术的区域性多源地学空间信息集成若干问题探讨[J]. 地球科学——中国地质大学学报,2001, 25(6): 545—550.
LI C L, ZHANG K X. Study on regional multi-source geological spatial information system based on techniques of GIS [J]. Earth Science—Journal of China University of Geosciences, 2001, 25(6): 545—550.
- [2] 周成虎,李军. 地球空间元数据研究[J]. 地球科学——中国地质大学学报,2000, 25(6): 579—585.
ZHOU C H, LI J. Research into geo-spatial metadata [J]. Earth Science—Journal of China University of Geosciences, 2000, 25(6): 579—585.
- [3] 袁艳斌,吴冲龙,李伟忠. 面向野外地质填图的空间实体对象表达[J]. 地球科学——中国地质大学学报,2001, 26(2): 192—196.
YUAN Y B, WU C L, LI W Z. Object styles of spatial entity for field geological mapping based on object-oriented method [J]. Earth Science—Journal of China University of Geosciences, 2001, 26(2): 192—196.
- [4] 张夏林,汪新庆,吴冲龙. 计算机辅助地质填图属性数据采集子系统的动态数据模型[J]. 地球科学——中国地质大学学报,2001, 26(2): 201—204.
ZHANG X L, WANG X Q, WU C L. Dynamic data model applied in regional geological mapping computer-aided mapping system [J]. Earth Science—Journal of China University of Geosciences, 2001, 26(2): 201—204.
- [5] 李昭原. 数据库技术最新进展[M]. 北京:清华大学出版社,1997.
LI Z Y. Proceeding of database technology [M]. Beijing: Tsinghua University Press, 1997.

Object-Relation Model of ORACLE Spatial Database: with Discussion of RDB and OODB Application in GIS

LI Zhen-hua

(Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China)

Abstract: There are some problems of RDB in GIS application. Sometimes the data type is too simple to describe non-structured or complex-structured data in GIS and the simple model can't express the relation of hierarchy, inheritance, aggregation, generalization or specialization, or there is impedance mismatch between SQL and programming language, especially the current popular object-oriented language. While object-oriented (OO) technology may be useful to solve these problems. ORACLE spatial database extends the traditional relation database with the technology, the so-called object-relation model, which features adding to RDB an object data type named SDO_GEOMETRY to store geometry. An application example of ORACLE spatial database demonstrated the advantages of OODB in describing spatial data and the efficiency in expressing compound object and processing spatial data compared to RDB which only stores meaningless data. Certainly, there are still some issues that need further research in OODB: which direction OODB will develop; pure OODB or object-relation DB, the data model match between RDB and OODB, the mathematic basis and formalization for OODB and how to take advantages of existing mature technologies of RDB such as transaction control, integrity check, data reproduction in the development of OODB.

Key words: OODB (object-oriented database); spatial database; RDB (relational database); GIS (geographic information system); ORACLE; object-relation model.