Guide to Developing & Application With Oracle Spatial Database

Oracle Spatial

空间数据库

开发应用指南

何原荣
李全杰 编著
傅文杰

测绘出版社

Oracle Spatial 空间数据库开发应用指南

Guide to Developing & Application With Oracle Spatial Database

何原荣　李全杰　傅文杰　编著

测绘出版社

·北京·

内 容 简 介

本书以 Oracle 10g Spatial 为范本，深入讲解了对象-关系模式的 Oracle Spatial。内容主要包括空间索引与空间查询、空间数据（包括矢量数据与栅格数据）的存储与管理、空间坐标系、拓扑数据模型、网络数据模型、栅格数据模型等理论，还包括使用 Java API、PL/SQL API，以及从 Oracle 9i 开始提供的开发方式 OCCI 如何开发基于 Oracle Spatial 的应用程序。最后在本书的附录中解析了 Oracle 10g Spatial 几乎所有的包、函数与过程。

本书的读者对象包括从事 GIS、CAD、CAM、RS 行业的工作者，高校中 GIS 专业、交通专业、环境专业、测绘专业等有空间数据库课程的本科生、硕士生、博士生，深入学习 Oracle 数据库的读者，深入学习对象-关系数据库的读者以及正在使用 Oracle Spatial 的用户。也可以作为空间数据库课程的参考用书。

本书各章节示例的源代码，如读者需要可与作者联系。E－mail：liquangjie@glite.edu.cn

图书在版编目(CIP)数据

Oracle Spatial 空间数据库开发应用指南/何原荣，李全杰，傅文杰编著．—北京：测绘出版社，2008.6
ISBN 978-7-5030-1861-9

Ⅰ.O... Ⅱ.①何...②李...③傅... Ⅲ.关系数据库-数据库管理系统，Oracle－程序设计 Ⅳ.TP311.138

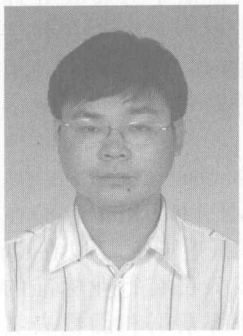
中国版本图书馆 CIP 数据核字(2008)第 076849 号

责任编辑 金 君　　封面设计 杨晓明

出版发行 测绘出版社
社　　址 北京西城区复外三里河 50 号　　邮政编码 100045
电　　话 010－68512386　68531609　　网　　址 www.sinomaps.com
印　　刷 北京市通州次渠印刷厂　　经　　销 新华书店
成品规格 184mm×260mm　　印　　张 22.5
版　　次 2008 年 6 月第 1 版　　印　　次 2008 年 6 月第 1 次印刷
印　　数 0001－1500　　定　　价 46.00 元

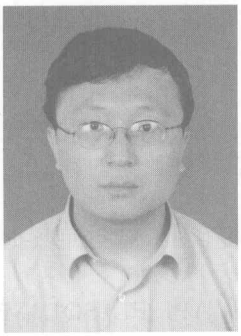
书　　号 ISBN 978-7-5030-1861-9/P・480

作者简介



何原荣

男，汉族，中南大学地图制图学与地理信息工程博士研究生，1977年8月生，福建省龙海市人，现任教于桂林旅游高等专科学校，主要从事资源环境GIS与遥感应用研究，主持完成省部级项目1项，参与国际合作项目1项、省厅级项目8项，发表研究论文10余篇，参编教材2部。



李全杰

男，汉族，桂林工学院地图制图学与地理信息工程硕士研究生，1978年3月生，河南商丘人，任教于桂林工学院，主要从事WebGIS、Grid GIS的研究。

傅文杰

男，汉族，1967年2月生，福建省莆田市人，2006年6月获中南大学国土资源信息工程专业博士学位，高级工程师，现任教于莆田学院环境与生命科学系，主要从事国土资源信息工程与3S应用研究，主持完成省部级项目2项，参与国家“十五”攻关科技项目1项、省部级项目5项，获得部级科技进步一等奖1项，2007年被评为“福建省新建本科院校跨世纪人才”。已在核心刊物上发表研究论文10多篇。

前　言

为满足多媒体和地理信息系统等非传统关系数据库应用的需求，Oracle 公司提供了空间数据暗盒(Spatial Cartridge，SC)用来存储和管理空间数据。从 Oracle7. 3. 3 版本就可以使用 SC 了。SC 也就是 Oracle Spatial 的关系模式，是标准的 C/S 模式产品，提供了分布式处理能力，全部的 Oracle 安全管理机制，SQL 方式访问空间数据等功能。SC 的空间数据管理和空间数据分析完全由 Oracle 来执行，对图形数据的管理采用的是一种开放的方式，任何授权用户可以通过 SQL 数据管理语言(DML)进行空间数据的添加和删除等操作；SC 采用多记录多字段存储空间数据。

随着 Oracle 8i 的推出，Oracle 公司把 SC 升级为 Oracle Spatial。在 Oracle Spatial 中，引入了 SDO_GEOMETRY 等抽象数据类型来表示空间数据，SDO_GEOMETRY 可以存储在一个字段中，与普通的自定义类型没有什么区别，可以作为字段的类型使用，这样就可以在 Oracle 数据库中快速有效地存储、访问和分析空间数据。也方便了应用开发人员在行业标准的数据库服务器中存储所有位置(地理位置)信息，而无须再求助于定制的外部索引和函数去实现他们所需要的此类功能。

Oracle Spatial 提供对象-关系模式和关系模式两种方式来存储空间数据。前者的特征是空间表中有一个类型为 MDSYS. SDO_GEOMETRY 的字段，后者也就是空间数据暗盒，其特征是每一个空间几何图层对应 4 个表，分别为_SDOLAYER、_SDODIM、_SDOGEOM、_SDOINDEX。这些表并不包括属性数据，属性数据需建立连接。

本书深入讲解对象-关系模式的 Oracle Spatial。从 Spatial 的版本上来看，本书不仅涉及从 Oracle 7. 3. 3 开始到 Oracle 8. 1. 6 之间的 SC，而且还涉及 Oracle 8i、9i Spatial 以及最近推出的 Oracle 10g Spatial，并以 Oracle 10g Spatial 为依据。内容主要包括矢量数据的存储与管理、空间数据的上载、空间坐标系。

拓扑数据模型、网络数据模型、栅格数据模型是 Oracle 10g Spatial 提供的，对空间数据的存储和管理提供了较为完备的支持。

本书还讲解如何开发基于 Oracle Spatial 的应用程序，包括使用 Java API、PL/SQL API，以及从 Oracle 9i 开始提供的开发方式 OCCI 等。

从功能的角度看，Oracle Spatial 是一组包(packages)、函数(function)与过程(procedure)，比如 SDO_GEOM 包、SDO_CS 包以及 SDO_FILTER、SDO_RELOATE、SDO_NN 等函数与过程。通过这些包、函数和过程，能够把空间数据(包括矢量数据和栅格数据)存储进 Oracle 数据库中，并能够对空间数据进行高效快速地访问，对空间数据进行空间分析。因此，在本书的附录部分，详细解释了 Oracle Spatial 的各个包。

空间数据库的另两个重要部分是空间索引和空间查询语言。当前比较常用的空间索引有四叉树索引和 R-树索引。本书还讲解 Oracle 10g Spatial 的空间索引与空间查询。

对象-关系数据库是下一代主流数据库，Oracle 是典型的对象-关系型数据库，Oracle Spatial 就是 Oracle 对象-关系模式的一个应用。本书还从对象-关系数据库的角度阐述 Oracle Spatial，是深入学习对象-关系型数据库读者的一本不可多得的理论与使用相结合的书。

本书第 1、2、3、5 章由何原荣编写，第 4、9 章及附录由李全杰编写，第 6、7、8 章由傅文杰编写，最终由何原荣负责统稿完成。鉴于编者的能力和时间限制，加上对象关系空间数据库技术目前尚处于快速的发展之中，偏颇和不足之处敬请读者不吝提出批评指正，以便修订再版时改进。

编者

2008 年 3 月

目　录

第 1 章　Oracle Spatial 概述

数据库发展至今，已经经历了三代，第一代是网状与层次数据库、第二代是关系数据库，第三代是对象-关系型数据库。空间数据库属于第三代数据库系统，随着地理信息系统(Geographic Information System，GIS)的开发和应用而发展起来的数据库。

GIS 空间数据的存储和管理大致经过了纯文件方式管理图形数据与属性数据、图形数据文件方式管理与属性数据关系型数据库管理、空间数据与属性数据一体化的管理方式三个阶段。目前，大多数 GIS 软件都倾向于第三种管理方式，即图形数据与属性数据都采用数据库管理的方式。针对这种需求，大型数据库厂商都相继推出了自己的数据库空间选件，如 Oracle 的 Oracle Spatial、IBM 的 DB2 Spatial Extender 以及 Informix Spatial DataBlade(目前 Informix 已被 IBM 收购)。

§1.1　Oracle Spatial

本节从 Oracle Spatial 的定义、两种模式、空间索引、空间查询 4 个方面理解 Oracle Spatial。

1.1.1　什么是 Oracle Spatial

Oracle Spatial 简称 Spatial，是 Oracle 公司推出的空间数据库选件，用来在 Oracle 数据库系统中存储和管理空间数据。

从功能角度看，Spatial 是一组包(Packages)、函数(Function)、过程(Procedure)，如 SDO_GEOM 包、SDO_CS 包与 SDO_FILTER、SDO_RELOATE、SDO_NN 等函数与过程。通过这些包、函数和过程，能够把空间数据(包括矢量数据和栅格数据)存储进 Oracle 数据库中，并对空间数据高效快速地访问以及进行空间分析。在本书的附录部分，详细解释了 Oracle Spatial 的各个包、函数与过程。

从总体来看，Spatial 又是一个集成化的环境，它给空间数据提供了一个 SQL 模式(MDSYS)以及该模式的一组对象数据类型(如 SDO_GEOMETRY，SDO_TOPO_GEOMETRY 等)，并给这些对象数据类型提供了索引、函数与过程，这些函数和过程把空间数据更容易地存储在 Oracle 数据库里，并能够高效快速地访问和分析。

总之，Oracle Spatial 包含以下几个方面的内容：

(1)在 DBMS 中存储空间矢量数据与栅格数据的能力，为用户提供了方便的空间支持。如 SDO_GEOMETRY 数据类型用来存储矢量数据、SDO_TOPO_GEOMETRY 用来存储拓扑数据、SDO_GEORASTER 与 SDO_RASTER 存储栅格数据等。

(2)提供空间索引机制，包括四叉树索引与 R-树索引以及操作这两类索引的函数，如 ESTIMATE_TILING_LEVEL 用来估计定长四叉树索引时四叉树的深度。

(3)提供用来规定 Oracle 支持空间数据类型的存储、语法、语义的模式(Schema)，即

MDSYS。

(4)摆脱了传统上空间数据与属性数据分开存储的GIS的数据管理混合体系结构,把空间数据与属性数据一体化存储,从而降低了系统管理的复杂性。

在表示公路的地图上,有代表城市、公路、国家和省的边界线,分别用点、线和多边形来表示,这些就是空间数据。它们表示了原有对象的地理位置(如经度、纬度或高度、深度),同时也保留了这些对象的相对位置和相对距离。地理信息系统的空间数据库就是用来存储、检索和表达这些空间数据的数据库。当然,对于一幅地图来说,除了这些空间数据以外,还有一些属性数据,如街道和省、国家的名字等。存储、检索、更新和查询空间数据和属性数据就是Oracle Spatial的主要任务。

计算机辅助设计(CAD)和计算机辅助制造(CAM)的空间数据也可以使用Oracle Spatial来存储和管理。

1.1.2 两种模式及其特点

Oracle Spatial提供了对象-关系与纯关系两种模式来存储与管理空间数据。纯关系模式就是Spatial Cartridge(SC)。

1. 关系模式及特点

SC存储空间数据时,每个图层对应4张关系表格,名字分别为〈layername〉_SDOLAYER、〈layername〉_SDODIM、〈layername〉_SDOGEOM、〈layername〉_SDOINDEX,这些表并不包括属性数据,属性数据需要存储在另外的表格中,其中〈layername〉为图层的名字。

关系模式有如下特点:①允许对数据库进行复制操作;②支持分布式数据库;③支持对象的分割与索引的并行装载。如一个很长的线对象在一条记录中不能完全存储时,可以使用两条或多条记录来存储。

2. 对象-关系模式及其特点

了解Spatial的对象-关系模式之前先了解对象-关系型数据库。对象-关系数据库是当前的主流数据库,是扩展关系数据库的数据库。Oracle从8i开始就是一个对象-关系型数据库。

(1)对象-关系数据库

面向对象数据库管理系统(OODBMS)必须满足两个条件[1]:①支持一个核心的面向对象数据模型;②支持传统数据库系统所有的数据库特征。也就是说OODBMS必须保持第二代数据库系统已有的技术。OODBMS不仅能很好地支持对象管理和规则管理,而且能更好的支持原有的数据管理。对象-关系数据库系统就是按照这样的目标将关系数据库系统与面向对象数据库系统两方面的特征相结合。

对象-关系数据库除了具有关系数据库的各种特点以外,还有其自身的特点:

①扩充数据类型

无论是网状与层次数据库,还是关系型数据库管理系统,它们只支持固定的数据类型集,如字符型、整型等,不能依据某一应用所需的特定数据类型进行扩展出新的类型集。对象-关系数据库则允许用户根据应用需求定义自己的数据类型、函数和操作符,对原有数据类型进行扩充。在Oracle中,纯粹的关系模式中不能表达一个3维向量对应的数据类型,因为在数据库中没有一个数据类型能包含三个实数;而在对象-关系数据库中,就可以定义这样的对象类型。如在Oracle数据库中就可以定义一个表示3维向量的对象类型,即

```
create or replace type Vector as object(
x      number,
y      number,
z      number);
```

②支持复杂对象

对象-关系数据库中允许在 SQL 中支持复杂对象。复杂对象是指有多种数据类型或用户自定义的数据类型构成的对象。在 Oracle 数据库中创建包含复杂对象的表格，如

```
create table Vec(
  name   varchar2(32),
  Vec    vector);
```

③支持继承的操作

能够支持子类、超类、继承的概念，包括属性数据的继承和函数及过程的继承；支持单继承与多重继承；支持函数重载、操作的重载。

④提供通用的规则系统

对象-关系数据库允许提供强大而通用的规则系统。规则在 DBMS 中是十分重要的，在传统的 RDBMS 中用触发器来保证数据库数据的完整性。触发器可以看成规则的一种形式。对象-关系数据库系统要支持的规则系统将更加通用，更加灵活，并且与其他的对象-关系能力是集成为一体的，例如规则中的事件和动作可以是任意的 SQL 语句，可以使用用户定义的函数，规则能够被继承等。

(2)Oracle 数据库的对象-关系模式

Oracle 的对象-关系模式是在关系模式的基础上扩展的，是关系模式之上的一个逻辑层，对象类型仍然支持原来的 SQL 接口，如查询(SELECT... FROM... WHERE)、快速提交、备份与恢复、行级锁定、分区表、并行查询、导入导出(exp、imp)、装载(SQL * Loader)等。引入了对象类型以后，PL/SQL、Java、OCI(Oracle Called Interface)、Pro * C/C++与 OO4O 等就有了新的编程接口，结果就导致了对象-关系模式的产生。

对象-关系模式的对象类型类似于 C++与 Java 的类，可以像使用类一样使用对象类型来对现实世界进行建模，使数据库应用程序的开发变得更容易、效率更高。对象类型存储在数据库服务器端，允许客户端应用程序直接访问，而不用通过客户端的对象到服务器端的关系映射。

①对象类型

对象类型是一种数据类型，如 Oracle Spatial 的类型 SDO_TOPO_GEOMETRY。该类型可以像使用 NUMBER、VARCHAR2 等基本数据类型一样来使用，可以把表中的字段类型声明为 SDO_TOPO_GEOMETRY 类型，也可以把 PL/SQL 的变量声明为对象类型。对象类型的值就称为该对象类型的一个实例，一个实例也叫一个对象。在 Oracle 数据库中创建了一个名为 person_typ 的对象类型，如

```
CREATE TYPE person_typ AS OBJECT (
    idno       NUMBER,
    name       VARCHAR2(30),
    phone      VARCHAR2(20),
```

```
    MAP MEMBER FUNCTION get_idno RETURN NUMBER );
/

CREATE TYPE BODY person_typ AS
  MAP MEMBER FUNCTION get_idno RETURN NUMBER IS
    BEGIN
      RETURN idno;
    END;
END;
/
```

对象类型与普通数据类型也有一些不同的特点：

——数据库中现成的定义好的对象类型很少，所以用户需要时，还要自己定义。

——对象类型由属性与方法组成。

·属性用来存储数据，例如，一个 student 对象类型可能包含姓名、专业、毕业日期等属性。属性的数据类型既可以是基本数据类型，又可以是定义的对象类型。属性用来存储对象的实例数据。

·方法定义了对象的行为，是用来处理对象的数据的过程和函数。Oracle 对象类型的方法不是必需的，一个对象类型可以有方法也可以没有方法。

·对象类型比普通的数据类型更容易模拟现实世界，这也是对象类型的优点。用户可以定义与现实世界对应的对象类型，如定义应用程序需要处理的顾客类型或者订单对象类型，这些数据类型使得在处理数据时的方法变得更容易。

②对象

对象是对象类型的实例，也可以说是对象类型的一个变量(不过在该变量中存储了数据)。对象具有对象类型的属性与方法，使用对象就是使用对象的属性与方法。

定义了对象类型以后，就可以像使用 NUMBER、VARCHAR2 等数据类型一样来使用，如下面的例子中就使用了刚才创建的数据类型。

首先创建表格：

```
CREATE TABLE contacts (
  contact        person_typ,
  contact_date   DATE );
```

然后向表中存储数据：

```
INSERT INTO contacts VALUES (
    person_typ (1, 'Jet Lee', '13000000010'),'10-3 月-04' );
```

contacts 就是使用了对象类型创建的关系表，具有对象类型的列叫做对象列。

对象的值可以有空(NULL)，如在上面的例子中，可以给对象的值赋空(NULL)，有两种方法，即

```
INSERT INTO contacts VALUES (NULL,'11-3 月-04' );
```

或者

```
INSERT INTO contacts VALUES (person_typ (NULL, NULL, NULL),'12-3 月-04');
```

第一种情况下 Oracle 直接把 contact 字段设置为 NULL，不会给类型 person_typ 的对象分配空间；第二种情况下，Oracle 会给类型 person_typ 的对象分配空间，但是该对象的所有属性都赋值为 NULL。

③方法与属性

对象类型的属性是基本的数据类型或其他对象类型，是对象类型的数据部分，方法则是组成对象类型的 PL/SQL 子程序，是对象类型的行为部分。例如，数据类型 person_typ 中，idno、name 与 phone 就是对象类型的属性，get_idno 是对象类型的方法。

定义属性时必须提供属性名与属性数据类型，Oracle 的大部分数据类型都可以作为属性类型，但是 LONG 和 LONG RAW、ROWID 和 UROWID、PL/SQL 特有类型，如 BINARY_INTEGER，BOOLEAN，%TYPE%，%ROWTYPE，REF CUSOR，RECORD，OLS_INTEGER 等类型除外。

注意：定义属性时既不能指定对象属性的默认值，也不能指定为 NOT NULL。

方法就是函数与过程，用于描述对象要执行的动作。Oracle 的对象类型的方法有 4 种，这里只作粗略的介绍。

Member：用于访问对象实例数据。这种类型的方法既可以是函数，也可以是过程。例如，

```
create or replace type person as object(
    name    varchar2(32),
    address varchar2(255),
member procedure edit_address(new_address varchar2)
);
/

create or replace type body person is
  member procedure edit_address(new_address varchar2) is
  begin
    address:= new_address;
  end;
end;
/
```

基于类型 person 执行下面操作：

```
create table doctor(
  no           number(6),
  person_obj   person);
insert into doctor values (1,person('何原荣','中南大学'));
insert into doctor values(2,person('李全杰','桂林工学院'));
insert into doctor values (3,person('傅文杰','莆田学院'));
```

使用 member 方法：

```
SQL> declare
  2  v_person person;
```

```
3  begin
4  select person_obj into v_person from doctor where no=&&no;
5  v_person.edit_address('桂林市建干路12号桂林工学院');
6  update doctor set person_obj= v_person where no=&no;
7  end;
8  /
Enter value for no: 1
old  4: select person_obj into v_person from doctor where no=&&no;
new  4: select person_obj into v_person from doctor where no=1;
old  6: update doctor set person_obj= v_person where no=&no;
new  6: update doctor set person_obj= v_person where no=1;

PL/SQL procedure successfully completed.
```

更新成功以后就可以使用SELECT查询表的内容了。

Static:这种类型的方法用于访问对象类型,在对象类型上执行全局操作。读者需要特别注意,Static类型的方法只能由对象类型调用,而不能由对象调用。

Map:这种方法是对象类型的一种可选方法,可以将对象映射为数据,如映射为Date、Number或Number2等,然后就可以对数据执行排序等操作。但要注意,每一个对象类型只能有一个Map方法。

Order:类似于Map方法,可以把对象映射为普通的数据,然后进行排序,但是这种类型的方法只能映射两个对象之间排序,同时一个对象类型只能定义一个这种类型的方法。

方法提供了用于访问对象数据的一种途径,在应用程序中需要调用这种操作时,可以直接调用这些方法,而不用再重新定义。例如:

```
SELECT c.contact.get_idno() FROM contacts c;
```

注意:必须使用别名才能调用方法,而不能直接使用列名。

④继承

继承和多态是面向对象技术的主要特点。Oracle对象的继承与C++或Java的继承相似,子对象除了具有父对象的所有特征,还添加了新的属性和方法。

子对象继承父对象是通过把父对象作为一个属性来定义的。在下面的例子中,创建的数据类型student_typ就继承了父对象person_typ。

```
CREATE TYPE student_typ AS OBJECT (
  School   varchar2(32),
  Person   person_typ);
```

⑤对象表

对象表是一种特殊的表,它的每一行都是一个对象。例如创建一个对象表:

```
CREATE TABLE student_obj_table OF person_typ;
```

有两种方法来处理该表,①每一行作为一个对象来处理,这样可以执行该对象的方法。②把类型person_typ的属性idno、name与phone作为字段来处理,这样就可以执行一些关系操作。可以执行下面的命令:

INSERT INTO student_obj_table VALUES (1, 'Jet Lee', '13000000010');

SELECT VALUE(s) FROM student_obj_table WHERE s.name = 'Jet Lee';

第一条语句把表 student_obj_table 作为一个多字段的表格，向该表中存储一条记录。第二条语句把 student_obj_table 作为单字段表格，并使用 VALUE 函数把返回值作为对象。缺省的情况下，对象表中的每个对象都有唯一的对象 ID(OID)。

⑥行对象与列对象

行对象指对象表中的每一个对象，因为每个对象都占用一行，所以叫做行对象。列对象是指一条记录中数据类型为对象类型的字段中存储的对象，使用对象-关系模式的 Spatial 时，存储空间信息的类型为 MDSYS.SDO_GEOMETRY 的字段就是列对象。

⑦REF 数据类型

REF 是指向行对象的指针，是 Oracle 的一种内置数据类型。创建表格时使用 REF 类型引用行对象，可以使不同的表共享相同的对象，从而降低内存占用。

(3)Oracle Spatial 的对象关系模式及其特点

使用对象-关系模式存储空间数据时，每个图层对应一个 2 维表格，并且图层中的每个图元对应着表中的一行。该表含有一个类型为 MDSYS.SDO_GEOMETRY 的列，图元的空间属性就存储在该列中。

对象-关系模式的特点：

①对象-关系模式支持的空间数据类型除了基本的数据类型以外还有：弧、圆、组合多边形(多边形的边界既有直线边又有弧边)、组合直线(由直线与弧组成的对象)、矩形等。

②使用对象-关系模式更容易建立空间索引，更方便执行空间查询。空间索引的创建和维护由基本 SQL 的 DDL(Create、Alter、Drop)与 DML(Insert、Update、Delete)完成。

③对象-关系模式的空间索引机制由服务器维护，不需要用户参与，降低了出错的机会。在关系模式中用户需要自己维护〈layername〉_SDOINDEX 索引表，包括索引表的创建与索引的生成。

④对象-关系模式中，空间对象存储在每行类型为 MDSYS.SDO_GEOMETRY 的字段中(一张表可以有多个这种类型的字段)。

⑤Oracle Spatial 在对象-关系模式中，加强了对空间数据的存储和查询性能。

由于 Oracle Spatial 是遵循 Open GIS 规范的产品，所以这两种模式都严格遵循 Open GIS ODBC/SQL 规范。对象-关系模式的接口是标准 SQL 的空间扩展——操作空间数据类型的 SQL，而关系模式的接口则是标准 SQL。

在不使用 Oracle 数据库分布式特性的情况下，为了有更高的效率，Oracle Spatial 建议最好使用对象-关系模式来存储空间数据。

1.1.3　空间索引

在 Oracle 10g Spatial 日常操作中，空间查询占绝大部分。因此，提高空间查询的性能，也就相应可以提高空间数据库的性能。通常有两种方法用于提高其性能，一种是优化查询语句，另一种是在将要查询空间对象的表格的查询字段上建立空间索引。对空间数据的查询处理是一项时间和空间开销都很高的操作，为了有效提高对空间数据的处理效率，尤其是实时查询效率，仅仅对查询语句进行优化是远远不够的，数据库管理员必须给数据库建立有效的空间索引

机制。在 Oracle 10g Spatial 中，相同的查询不同的空间索引得到的查询结果却不同，差的空间索引可能会引起 I/O“瓶颈”，而合适的空间索引则可以大大提高空间查询的效率，能否设计出合适的空间索引，可以作为判断 Oracle 空间数据库管理员管理水平高低的标准。

1. 空间索引的构建过程

空间索引的基本原理相似，都采用分割原理，即把查询空间划分为若干区域（通常为矩形或多边形），这些区域或单元包含空间数据并可唯一标识。目前有两种分割方法，一种是规则分割方法，另一种是基于对象的分割方法。规则分割方法是将地理空间按照规则或半规则方式分割，分割单元间接地与地理对象相关联，地理要素的几何部分可能被分割到几个相邻的单元中，这时地理对象的描述保持完整，而空间索引单元只存储对象的位置参考信息。在基于对象的分割方法中，索引空间的分割直接由地理对象来确定，索引单元包括地理对象的最小外接矩形。

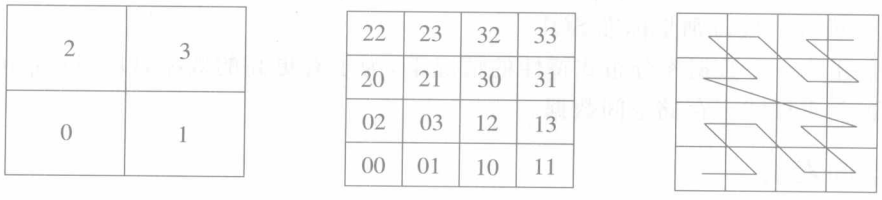
与前面的索引划分原理相对应，有两类常用的空间索引，分别为 R-树系列（对应基于对象分割方法）和四叉树系列（对应规则分割方法）。目前国内外主要的空间数据库大都采用这两类空间索引方法，如 ESRI 的 ArcView 、MapInfo 公司的 MapInfo 和 Informix 的 GeoSpatial DataBlade 采用的是 R-树索引作为空间索引。中国地质大学的 MapGIS 和中科院的 SuperMap 则采用的是四叉树索引作为空间索引，而 Oracle Spatial 则同时引入 R-树索引和四叉树索引这两类索引。

2. 四叉树索引

Oracle 10g Spatial 空间索引是逻辑索引，索引的入口点就是空间对象在坐标系中的坐标（即查询时所查询的索引数据），也就是索引数据表中相应的记录（而索引的对象却可以在空间中的不同的小区域里）。Oracle 10g Spatial 四叉树索引分定长（fixed indexing）索引与变长索引（hybrid indexing）。

（1）定长索引

图 1-1(a)把整个地理空间划分成四个等份空间，图 1-1(b)继续把每个空间四等分，如此持续递归划分，直到满足条件为止，生成的网格按照图 1-1(c)的顺序编码，称为 Z 顺序码或莫顿码（Morton），生成的四叉树是满四叉树，对应的索引称为定长索引。在数据库中把这些莫顿码按一定顺序存储进表格中，该表称为索引数据表。空间索引数据表的生成过程也就是空间索引的创建过程。执行查询时，首先按照空间对象的大致大小在索引数据表中按照一定的搜查找法进行查找，如快速查找、折半查找等，直至找到目标对象。执行过程如图 1-2 所示。



(a)深度为 1 的四叉树　(b)深度为 2 的四叉树　(c)四叉树编码顺序

图 1-1　四叉树的划分与莫顿编码