

Oracle Spatial 最佳实践

Oracle 技术白皮书
2003 年 12 月

目录

1 概述	1
2 数据建模	2
3 元数据、容限和坐标系统	2
4 数据加载	4
5 数据验证	4
6 为空间数据创建索引	5
7 分区的空间索引	6
8 空间查询	6
9 应用程序考虑事项	8

Oracle Spatial 和 Oracle Locator 是 Oracle 数据库强大的核心特性。本技术文档介绍了一些最佳实践、提示和一般信息,它们能够帮助您利用 Oracle Spatial 和 Oracle Locator 在您日常的业务实践中提高生产效率、改进决策支持和促进成本节省。

Oracle 数据库 10g 中的 Oracle Spatial 包含了用于存储矢量数据类型、栅格数据类型和持续拓扑数据的原生数据类型。本文档概述了在 Oracle 数据库 9i 和 10g 的 Oracle Spatial 中使用 Oracle 的原生矢量数据类型 SDO_GEOMETRY 的一些最佳实践。

1 概述

Oracle Locator 与 Oracle 的标准版和企业版捆绑在一起。Locator 是 Spatial 的一个子集。获得 Oracle 标准版或企业版的许可将使您无需另外付费即可完全享用 Oracle Locator 中提供的宝贵特性集。

SDO_GEOMETRY 是在 Oracle 中提供的唯一矢量数据类型,它自始就集成到了所有主要 GIS 供应商的产品套件中。

一些主要的 GIS 供应商可以使用旧的专用数据类型在 Oracle 中存储矢量数据,例如,使用 Oracle 的 LONG RAW 数据类型。这些旧数据类型的引入要早于 SDO_GEOMETRY 数据类型。在市场中,有时存在这样的误解,认为使用 SDO_GEOMETRY 作为基本矢量数据存储类型可能会危及主要 GIS 供应商所提供产品中的特性或性能。事实上,SDO_GEOMETRY 数据类型不会危及主要 GIS 供应商所提供产品中的任何出色特性。如果您遵照本文所述的准则,那么最好的替代专用数据存储类型与 SDO_GEOMETRY 数据类型在性能上的偏差绝不会超过 +/- 15%。

实际上,如果您使用核心的 Oracle 特性(如表分区),那么 Oracle Spatial 的性能可能比专用的“LONG RAW”数据类型解决方案的性能远远高出 15%。这是因为 Oracle 不支持在包含 LONG 或 LONG RAW 列的表上进行表分区。具有 SDO_GEOMETRY 列的表可以利用 Oracle 表分区,这可以显著地帮助提高性能、可伸缩性和可管理性。

Oracle Spatial 可以利用 Oracle 数据库的核心特性。通过利用 Oracle 的核心公用程序和特性,当前的 Oracle DBA 和开发人员可以在使用 Oracle 的空间技术时使他们现有的 Oracle 知识库最大化。如果您了解 Oracle 而从未听说过 Oracle Spatial,那么您已经了解了这个 Oracle 特性的 80% 以上。用于非空间数据的核心 Oracle 公用程序(import, export, sqlldr)同样也可用于空间数据。同样,核心的 Oracle 特性(如表分区和高级复制)都可以和 Oracle Spatial 一起使用。Oracle 的空间战略是使空间数据在您的机构中成为主流,并使您的企业现有的 Oracle 知识库最大化。

本白皮书重点介绍了一些最佳实践和提示，以帮助设计和开发利用 Oracle 空间技术的应用程序。本白皮书中的许多建议不是专门针对 Oracle Spatial 的，这进一步强调了在您的企业内部利用现有 Oracle 知识的能力。本白皮书其余部分包含以下各节：

- 数据建模
- 元数据、容限和坐标系统
- 数据加载
- 数据验证
- 为空间数据创建索引
- 分区的空间索引
- 空间查询
- 应用程序考虑事项

2 数据建模

传统的 RDBMS 数据模型概念适用于处理空间数据的情况。Oracle 支持许多传统的数据类型，包括为字符提供的 VARCHAR2、为日期提供的 DATE 类型、为数字提供的 NUMBER 类型，以及现在为存储空间特征坐标而提供的 SDO_GEOMETRY 数据类型。

在 Oracle 中没有诸如空间表之类的东西，只有包含一个或多个 SDO_GEOMETRY 列的普通 Oracle 表。当您创建规范化的表时，Oracle 建议在表中包含 SDO_GEOMETRY 列，其中表内的所有其他列与 SDO_GEOMETRY 列均存在一对一的关系。

看看下面这个为道路空间特征和河流空间特征建模的例子。道路信息可能包含车道数、街道地址范围等。河流信息可能包含盐度、最大深度等。即使它们都是线性特征，但因为道路信息与河流不相关，且河流信息也与道路不相关，所以建议不要将它们的坐标存储在表的同一个 SDO_GEOMETRY 列中。一个规范化的数据模型将把道路空间特征保存在 Roads 表中，而表中其他列则与道路坐标存在一对一的关系。而且，推荐将一个类似的规范化数据模型用于 Rivers 表。

在查询时您将更深刻地体会到将道路信息与河流信息分开存储的另一个好处。当您只搜索道路信息时，您就无需从既包含道路条目又包含河流条目的表中进行筛选了。

3 元数据、容限和坐标系统

表中的每一个 SDO_GEOMETRY 列都需要在 Oracle 空间元数据字典 USER_SDO_GEOM_METADATA 中有一个条目。元数据条目包含以下信息：

- 包含类型 SDO_GEOMETRY 的列的表的名称
- 用 SDO_GEOMETRY 数据类型定义的列的名称
- SDO_GEOMETRY 列的轴（维度）数
- 每一个轴的上下界
- 每个轴的容限值，一般所有轴的该值相同
- 空间参照标识符 (SRID)

每个轴的上下界不是 SDO_GEOMETRY 列中数据的最小限定矩形 (MBR)。轴边界应该是可以包含当前和未来的所有几何结构的数值。定义的第一个轴必须始终是 x，第二个轴是 y。此外，还可定义可选的 z 轴和测量轴。

当处理大地数据时（经度/纬度数据），第一个轴的范围必须定义为 (-180, 180)，第二个轴的范围定义为 (-90,90)。

x 轴和 y 轴的容限一般相同。容限是两个坐标被认为是同一个坐标时所不能超过的距离。Oracle 的几何验证例程、空间操作符和空间函数全都使用容限。定义一个反映您采集数据所用真实分辨率的容限非常重要。

当存储非经度/纬度数据时，容限单位与和空间数据相关的坐标系统单位相同。当存储经度/纬度数据时，容限单位是米。

Oracle Spatial 和 Oracle Locator 支持的所有坐标系统都定义在一个称为 MDSYS.CS_SRS 的字典表中。此外，还可以将定制的坐标系统添加到 MDSYS.CS_SRS 字典中，该过程在《Oracle Spatial 用户指南和参考》中进行了说明。在 MDSYS.CS_SRS 字典中，一个称为 SRID 的数字主键指出了每一个支持的坐标系统。字典表还包含以 Open GIS Consortium (OGC) 定义的通用文本 (WKT) 语法的形式表示的每个坐标系统的定义。将空间数据与坐标系统相关联，与将空间数据与 SRID 值相关联一样简单。

建议将空间数据与 SRID 相关联，特别是在您的数据是地理数据，即与地球相关的数据时。地理数据可以分为两类：大地数据（经度/纬度数据）和投影（非经度/纬度数据）。

Oracle 采用由大地 SRID 定义的几何结构连续坐标之间的大圆距离。

当把一个 SRID 与 SDO_GEOMETRY 列相关联时，必须在 USER_SDO_GEOM_METADATA 条目中、以及在加载的每个 SDO_GEOMETRY 对象的 SDO_SRID 属性中指定该 SRID。

4 数据加载

批量加载可以利用传统的 Oracle 公用程序（如 SQL*Loader 和 Import）来完成。批量卸载可以利用 Oracle 的 Export 公用程序来完成。这些公用程序不需要空间特有语法。如对非空间数据所建议的，如果您要执行一个大型批量加载，那么建议删除索引，包括空间索引（如果存在）、执行加载并在加载完成后重新创建索引。如果在批量加载前没有删除索引，那么在加载时将保留这些索引。

SQL*Loader 可以加载空间数据，但它不能识别地理信息系统（GIS）供应商的交换格式（如 ESRI shapefile、MapInfo Tab 文件、Autodesk DWG 文件或 Microstation DGN 文件）。每一个主要的 GIS 供应商都有它们自己的工具来将它们的交换格式导入成 Oracle 的 SDO_GEOMETRY 格式。也有通用的转换产品（如 Safe Software 提供的特征处理引擎（FME）），这些产品能够将许多供应商格式加载成 SDO_GEOMETRY 数据类型。FME 还可以提取存储在 Oracle 的 SDO_GEOMETRY 数据类型中的数据，然后将其转换成任意的 FME 支持的 GIS 供应商格式。

由于 ESRI shapefile 是一种非常常见的交换格式，因此 Oracle 在 Oracle 技术网上发布了一个免费的公用程序 shp2sdo。您可以从以下 URL 下载这个公用程序：<http://otn.oracle.com/products/spatial>。虽然 Oracle 不为该公用程序提供支持，但它已经经过了广泛的测试。之所以不为该工具提供支持，是因为它是用共享软件进行编码来读取 shapefile 的。shp2sdo 公用程序从一个形状文件中读取几何结构和属性。然后它创建：

- 一个 SQL 脚本，用以创建相应的 Oracle 表以及 SDO_GEOMETRY 列所需的相关元数据
- 一个控制文件，用于 Oracle Import 公用程序

如果您在使用 Oracle 的 shp2sdo 公用程序，那么在数据加载之后，建议您运行 SDO_MIGRATE.TO_CURRENT 过程将数据移植到 Oracle 推荐的最新格式。如果您在使用 GIS 供应商公用程序，或 Safe Software 的 FME 产品，则不需要运行 SDO_MIGRATE.TO_CURRENT。

5 数据验证

空间数据必须有效，以确保在您执行空间分析时得到正确的结果。如果为 SDO_GEOMETRY 列创建了空间索引，那么在将空间数据插入到列中时 Oracle 将执行一些有效性检查。但只有通过运行 SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT 或 SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT 过程，才会进行全面的验证。

如果在数据加载之前就可确保是有效的，则无需验证。否则，强烈建议进行验证。应当更正或删除无效的几何结构。

SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT 和 SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT 根据 Open GIS Consortium (OGC) 通过 SQL 简单特征规范定义的规则来验证几何结构。当报告无效的几何结构时（例如自交的多边形），还将报告其他相关信息（如哪些边相交了）。这些信息对于更正无效的几何结构非常有用。

所报告的一些最常见的验证错误包括：

可能报告的错误	原因及更正
ORA 13356 — 一个几何结构中邻近的重复点是冗余的。	容限可能设置得太粗略。将容限细化可能会修复这个错误。 点可能的确是重复的。删除重复的顶点。
ORA 13349 – 多边形边界自交。	容限可能设置得太粗略。将容限细化可能会修正这个错误。 多边形的确自交。通过确保没有边缘相交来修复这个多边形。
ORA 13367 — 内/外环旋转错误。	更正多边形环的旋转。外环应是逆时针的，内环应是顺时针的。

可以将验证例程报告的这些其他相关信息提供给以下例程来帮助修复无效的几何结构：

- SDO_UTIL.REMOVE_DUPLICATE_VERTICIES
- SDO_UTIL.EXTRACT

6 为空间数据创建索引

在 Oracle 8.1.7 中引入的 R-Tree 空间索引无需调整，并且我们建议将其用于几乎所有的情形。Oracle9i 引入了一个大地 R-tree 索引，它顾及了大圆距离以及跨越两极和 180 度子午线的几何结构。Oracle9i Release 2 在创建 R-tree 空间索引时引入了一定程度的并行化，并提供性能增强。Oracle 数据库 10g 引入了更多的 R-tree 性能增强、并发 DML 增强和更高的并行化。

在 CREATE INDEX 语句中指定 LAYER_GTYPE 参数将：

- 通过只允许将某类的空间特征插入到 SDO_GEOMETRY 列中来帮助保持空间完整性。例如，在创建索引时指定 LAYER_GTYPE=POINT 将只允许把点数据插入到列中。
- 帮助提高查询性能。如果在创建索引时指定了列的空间数据类别，则将调用空间查询优化。

以下 CREATE INDEX 参数与 R-tree 空间索引无关联，因此不应指定：

- SDO_LEVEL
- SDO_NUMTILES
- SDO_COMMIT_INTERVAL

7 分区的空间索引

在 Oracle8 中引入了 Oracle 表分区，以帮助提高包含许多行的 Oracle 表的性能和可管理性。在 Oracle9i 中引入了本地分区的空间索引，它能够为包含经过索引的 SDO_GEOMETRY 列的大型 Oracle 表提供更高的性能和可管理性。为了提高性能，请挑选一个大部分时间将在 WHERE 子句中出现的分区键。

如果您正在设计一个特大型数据库系统，并且可能考虑使用一个 GIS 供应商的旧专用 LONG RAW 空间数据存储格式，那么您可能需要三思了。Oracle 不支持包含任何 LONG 或 LONG RAW 列的分区表。当表变得越来越大时，这些解决方案在性能和可管理性方面无疑都将开始下降。

Oracle Spatial SDO_GEOMETRY 数据类型可以利用本地分区的空间索引的优势，以获得更快、本地化的搜索，以及可管理性特性（如分区级备份和交换）。

利用分区级的交换，位于分区表外的一个完整的表可以作为一个分区交换到分区表中。存在于仓库外表中的所有索引（包括空间索引）立即在分区表中变为可用。这是将新数据添加到仓库中的一种非常快速的方式，同时还使索引维护工作最小化。

8 空间查询

Oracle Spatial 包含一组空间操作符和函数。空间操作符和函数都执行空间分析，其差别在于操作符使用空间索引而函数不使用空间索引。空间操作符包括：

- SDO_FILTER (search_column, window, ‘操作符特有的参数’)
- SDO_RELATE (search_column, window, ‘操作符特有的参数’)
- SDO_WITHIN_DISTANCE (search_column, window, ‘操作符特有的参数’)
- SDO_NN (search_column, window, ‘操作符特有的参数’)
- 以及在 Oracle 数据库 10g 的 Spatial 中引入的其他操作符

空间函数包括：

- SDO_GEOM.SDO_AREA
- SDO_GEOM.SDO_LENGTH
- SDO_CS.TRANSFORM

- SDO_LRS.PROJECT_PT
- 以及许多其他函数

请注意，所有的空间操作符具有类似的参数标记：

- 空间操作符的第一个参数始终是被搜索的列，并且该列必须创建了空间索引。
- 第二个参数始终是查询窗口或感兴趣的区域。
- 第三个参数是包含了该操作符特有的参数列表的字符串。

当您编写包含空间操作符的 SQL 语句时，Oracle 提示可以帮助 Oracle 优化程序选择更好的执行计划。Oracle 提示不是 Spatial 特有的，但正如它们可以改善没有包含空间操作符的 SQL 语句的执行计划一样，它们也可以帮助改善包含空间操作符的 SQL 语句的执行计划。

为了获得最优的执行计划，在将多个几何结构传递给空间操作符的第二个参数时请始终指定 `/*+ ordered */` 提示。例如，以下查询查找 ID 值为 1 和 2 的被污染水井 5 英里内的所有化工厂。

```
SELECT /*+ ORDERED */
b.chemical_plant_name
FROM well_table a,
chemical_plants b
WHERE sdo_within_distance (b.geom, a.geom, 'distance=5 unit=mile') =
'TRUE'
AND a.id in (1,2);
```

在这个例子中，将多个水井位置（水井 ID 1 和 ID 2）传递给了操作符的第二个参数。这是一个典型的示例，在此应指定 `/*+ ordered */` 提示。当您指定了排序的提示时，以驱动表的顺序在 FROM 子句中列出各表。填充空间操作符第二个参数的表应在 FROM 子句中排在包含被搜索列的表前。在这个示例中，well_table 在 FROM 子句中排在 chemical_plants 之前。

空间操作符通过排除表中与操作符的第一个参数（搜索列）关联的行，缩小了备选结果的范围。当通过空间操作符缩小了查询结果的范围，并且 WHERE 子句中包含与填充空间操作符第一个参数相同的表的其他具有索引的列时，使用 `no_index` 提示可能会有所帮助。这在那些其他具有索引的列不是很具有选择性时尤为如此。

例如，假定修改了化工厂查询，以返回在 ID 值为 1 和 2 的受污染水井 5 英里内的“加工铬的”所有化工厂。假定 chemical_plants 表有一个称为 processes_chromium 的列，它的可能值为 ‘T’ 或 ‘F’（真或假）。即使 processes_chromium 列在其上有一个位图索引，它在查询中使用时可能不是非常有选择性的索引。在 processes_chromium 上提供一个 `no_index` 提示可以帮助 Oracle 优化程序避免有选择性的空间索引和无选择性的非空间索引的混合。

```
SELECT /*+ ORDERED
```

```
NO_INDEX (b processes_chromium_index_name) */
b.chemical_plant_name
FROM well_table a,
chemical_plants b
WHERE sdo_within_distance (b.geom, a.geom, 'distance=5 unit=mile') =
'TRUE'
AND a.id in (1,2)
AND processes_chromium = 'T';
```

Oracle 推荐进行基于成本的优化，并收集有关表和索引的统计信息。这可以利用以下过程实现：

- DBMS_STATS.GATHER_TABLE_STATISTICS
- DBMS_STATS.GATHER_SCHEMA_STATISTICS

包含 SDO_GEOMETRY 数据类型的表可以通过收集有关它的统计信息而获益，但不需要收集由 CREATE INDEX 命令隐式创建的空间索引表的统计信息。

9 应用程序考虑事项

如果可视化是您应用程序的一个主要部分，那么这部分的内容将非常适合您。当显示缩小得非常小时，打开非常详细的图层不是一种很好的做法。例如，如果缩小了显示来展示整个美国，那么提供详细的街道对显示没有任何裨益。在这种缩放级别上的详细街道在屏幕上看起来就像一个实点。在将显示放大到东西一千米范围时，提供详细的街道就比较逼真了。

下面是另一个示例。假定您有一个图层包含非常详细的多边形区域（每个多边形大约有 3000 个顶点）。当缩小得非常小时，显示这些非常复杂的多边形是没有意义的。由于这些多边形中的许多坐标被迫显示在仅仅几个象素点上，因此多边形的细节不见了。一个更实际的方法是使用缩放控制在放大到合理情况时才打开详细的多边形。另一种实际的方法是创建一个一般化图层（复杂多边形的一般化版本）。

当缩小得非常小时显示一般化图层，当放大到非常大时显示详细图层。

缩放控制和一般化图层的使用对于显示应用程序来说是众所周知的概念。正确使用缩放控制和一般化层将提供更好的性能。可以避免从服务器中不必要地读取详细的几何结构，特别是因为每个详细的几何结构的大多数坐标都将在仅仅几个象素点上显示。



版权所有 © 2003。甲骨文公司
保留所有权利

Oracle Spatial 最佳实践
作者：Daniel Geringer

Oracle Corporation 全球总部
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.
电话 650.506.7000
传真 650.506.7200

全球咨询：
电话 44.932.872.020
电报 851.927444(ORACLEG)
传真 44.932.874.625

<http://www.oracle.com>