

Oracle Locator 和 Oracle Spatial 11g 最佳实践

Oracle 技术白皮书
2008 年 8 月

Oracle Locator 和 Oracle Spatial 11g 最佳实践

引言.....	3
通用最佳实践.....	4
Oracle 补丁集.....	4
数据建模.....	4
元数据、容差和坐标系.....	4
使用空间数据和空间索引的可传输表空间.....	6
数据加载.....	6
SDO_GEOMETRY 对象的 SDO_POINT 属性中的点数据.....	6
几何结构验证.....	7
空间索引创建.....	8
分区的本地空间索引.....	9
空间查询.....	9
应用程序考虑事项.....	11
使用 ESRI ArcGIS 的最佳实践.....	12
DBTUNE 文件.....	13
关于 3D 数据的重要注意事项:.....	13
使用 ESRI 客户端工具创建空间索引.....	14
ArcSDE 的主键.....	14
非 ESRI 客户端工具创建的空间层的注册.....	14
从 ArcSDE 取消层的注册.....	14
使用 ESRI 客户端工具时的性能:.....	15

引言

本技术白皮书描述了在 Oracle Locator 和 Oracle Spatial 中使用用于存储矢量数据的 Oracle 数据库原生数据类型 SDO_GEOMETRY 的最佳实践，还提供了在 Oracle Locator 和 Oracle Spatial 上运行 ESRI ArcGIS 软件的指南。

Oracle Spatial 通过 Oracle Locator 对每个 Oracle 数据库中的核心位置特性进行了扩展。因此，在本文中介绍的 Oracle Locator 最佳实践也适用于 Oracle Spatial，而 Oracle Spatial 的实践只适用于 Oracle Spatial。

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 Locator 的，允许客户在企业内利用现有的 Oracle 知识。

Oracle Locator 受 GIS 和位置工具以及应用程序支持，并支持多种标准：所有主要的 GIS 和位置服务技术供应商以及地图绘制和图像数据提供商直接与 Oracle Locator 集成。他们支持原生的 Oracle SDO_GEOMETRY 数据类型，且特性和性能不受影响。Oracle 还始终如一地帮助制定、推动、实施和支持由开放地理空间联盟 (OGC) 提出的空间和位置服务领域的最新开放标准及 ISO SQL/MM 标准。

Oracle Locator 和 Oracle Spatial 是 Oracle 数据库原生特性。Oracle 数据库为包括空间数据在内的所有数据提供安全性、高可用性、容灾能力、可管理性和高性能。只有 Oracle 原生空间数据类型的用户才能充分利用所有这些特性：分区、本地分区的空间索引、包括空间索引的分区交换、支持空间数据和空间索引的可传输表空间、复制、工作区管理器（版本控制和有效时间）、并行索引构建和查询，以及以空间方式驱动的多级安全性。这些特性在使用 LONG RAW 或 BLOB 数据类型时不可用或功能受限。

Oracle DBA 和开发人员可以利用现有的 Oracle 知识，因为 Oracle 空间技术集成了这里和上一段列出的核心 Oracle 数据库公用程序和特性。如果您知道 Oracle 数据库，而从未听说过 Oracle Locator，那也已经了解这个 Oracle 特性的 80% 以上了。

用于非空间数据的核心 Oracle 公用程序 (impdp、expdp、import、export、sqlldr) 同样也可以用于空间数据。通过 Oracle, 用户可以使空间数据在您的组织内成为主流, 并利用现有的 Oracle 知识。

通用最佳实践

Oracle 补丁集

可能的话, 请升级到最新的 Oracle 数据库补丁集。这将确保您运行的 Oracle 数据库版本包含最新的空间特性和性能优化。

数据建模

传统的关系数据库数据模型概念适用于处理空间数据。Oracle 数据库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, 以及为存储空间特征坐标而提供的 SDO_GEOMETRY 数据类型。

Oracle 数据库没有空间表, 只有包含一个或多个 SDO_GEOMETRY 列的普通 Oracle 数据库表。当您创建规范化的表时, Oracle 建议在表中包含 SDO_GEOMETRY 列, 其中表内的所有其它列与 SDO_GEOMETRY 列均存在一对一关系。

考虑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。尽管它们都是线性特征, 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, 而表中其它列则与道路坐标存在一对一的关系。同理, 一个类似的规范化数据模型可用于 Rivers 表。

在查询时您将体会到将道路信息与河流信息分开存储的好处。当只搜索道路信息时, 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。

元数据、容差和坐标系

任何表中的每一个 SDO_GEOMETRY 列都需要在 Oracle Locator 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 SDO_GEOMETRY 类型列的表的名称
- 用 SDO_GEOMETRY 数据类型定义的列的名称
- SDO_GEOMETRY 列的轴 (维) 数
- 每一个轴的上下界

- 每个轴的容差值，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。当处理大地测量数据（经度/纬度数据）时，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` 轴的容差一般相同。容差是防止将两个坐标认为是同一坐标的距离。Oracle 的几何结构验证例程、空间操作符和空间函数全都使用容差。定义一个反映您采集数据所用真实分辨率的容差非常重要。

当存储非经度/纬度数据时，容差单位与和空间数据相关的坐标系单位相同。当存储经度/纬度数据时，容差单位为米。

Oracle Locator 支持的所有坐标系都定义在一个称为 `MDSYS.CS_SRS` 的字典表中。此外，Oracle 10g 第 2 版引入了欧洲石油调查组 (EPSG) 标准中代表坐标系的数据模型。每个 EPSG 定义的坐标系在 `MDSYS.CS_SRS` 中都有对应的条目。

Oracle Locator 预先配置有 1000 多个定义好的坐标系。这些预先配置的坐标系满足了众多用户的需求。如果这些预先配置的坐标系都无法满足您的要求，您可以添加自定义坐标系。

从 Oracle 数据库 10g 第 2 版开始，自定义坐标系必须通过在 Oracle 数据库中定义的 EPSG 数据模型输入。每个自定义的 EPSG 条目会自动在 `MDSYS.CS_SRS` 表中生成一个条目。添加自定义坐标系的流程详见《Oracle Spatial Users Guide and Reference》。

在 `MDSYS.CS_SRS` 字典中，一个称为 `SRID` 的数值型主键标识了每一个受支持的坐标系。该字典表还采用符合 Open GIS Consortium (OGC) 标准的通用文本 (WKT) 语法定义每个支持的坐标系。

将空间数据与坐标系相关联只需简单地将空间数据与 `SRID` 值相关联。

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：大地测量数据（经度/纬度数据）和投影数据（非经度/纬度数据）。Oracle 采用由大地测量 `SRID` 定义的几何结构连续坐标之间的大地测量距离。

含有空间数据和空间索引的可传输表空间

如果需将大量的空间数据和空间索引从一个数据库实例移动到另一个数据库实例，您可以考虑使用可传输表空间。

从 Oracle 数据库 10g 开始，只要源平台和目标平台的尾段格式相同，可传输表空间就支持传输空间索引。

只传输空间数据（不含空间索引）则不需要源平台和目标平台具有相同的尾段格式。

如果要传输表空间，您可能要重新考虑如何将数据和索引映射到表空间以帮助优化传输策略。

数据加载

批量加载可以利用传统的 Oracle 数据库公用程序（如 `sqlldr`、`imp` 或 `impdp`）来完成。批量卸载可以使用 Oracle 数据库导出公用程序 `exp` 或 `expdp` 等来实现。这些公用程序不需要空间任何特有语法。与对非空间数据的建议一样，如果您要执行大型批量加载或卸载，建议先删除索引（如果存在空间索引，还包括空间索引），执行批量事务处理，然后在事务处理完成后重新创建索引。如果在批量加载或卸载前没有删除索引，那么在事务处理时将保留这些索引。

`SQL*Loader` 可以加载空间数据，但它不能识别地理信息系统 (GIS) 供应商的数据格式（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，可以将它们的数据格式导入成 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，然后将其转换成任何 FME 支持的 GIS 供应商格式。

由于 ESRI shapefile 是一种非常常见的数据格式，因此 Oracle 在 Oracle 技术网上发布了一个免费的 Java 公用程序，用于加载 shapefile。您可以从以下 URL 下载这个公用程序：

<http://otn.oracle.com/global/cn/products/spatial>。这个公用程序是经过大量测试的优秀工具。它可以读取 shapefile 几何结构及其属性，并将记录加载（或附加）到 Oracle 表中。

SDO_GEOMETRY 对象的 SDO_POINT 属性中的点数据

如果可能，尽量将点数据存储到 `SDO_GEOMETRY` 对象的 `SDO_POINT` 属性中，而不要存储到 `SDO_ORDINATES` 属性中。

几何结构验证

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。如果一个 SDO_GEOMETRY 列上已有空间索引，那么在将空间数据插入到该列时，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 例程，才会进行全面的验证。

如果在数据在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 根据符合 Open GIS Consortium (OGC) 简单特征规范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），还将报告其它相关信息（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：

报告的错误	可能的原因及更正措施
ORA 13356 — 一个几何结构中邻近的重复点是冗余的。	容差可能设置得太粗略。将容差细化可能会修复这个错误。 点可能的确是重复的。删除重复点。
ORA 13349 — 多边形边界自交。	容差可能设置得太粗略。将容差细化可能会修正这个错误。 多边形确实自交。通过确保没有边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

SDO_UTIL 程序包还含有其它对检查几何结构有帮助的公用程序，例如：

- SDO_UTIL.GETNUMELEM – 返回几何结构中的元素数量
- SDO_UTIL.GETNUMVERTICES – 返回几何结构中的顶点数量
- SDO_UTIL.GETNUMRINGS – 返回几何结构中的环数量。

请参见相应文档了解更多公用程序。

空间索引创建

创建空间索引时，推荐使用以下参数。

- WORK_TABLESPACE – 在空间索引的创建过程中，该过程会创建中间表，索引创建完成后即删除这些中间表。中间表的大小最大为最终索引的两倍。如果没有指定 WORK_TABLESPACE，则会在最终索引所使用的同一表空间中创建中间表，这会导致碎片出现甚至性能下降。
可以使用 SDO_TUNE.ESTIMATE_RTREE_INDEX_SIZE，将其结果乘以 2 作为工作表空间大小的参考。可以重复使用同一工作表空间创建其它空间索引。
- LAYER_GTYPE – 此参数在处理仅包含点的层时尤为需要。如果仅包含点的层将它的点存储到 SDO_ORDINATE_ARRAY 中，您仍然可以在创建空间索引时指定 LAYER_GTYPE=POINT。这在执行空间分析时有助于提高查询性能。
- SDO_NON_LEAF_TBL – 此参数对于非常大的空间索引很有用（小型空间索引则不需要该参数）。该参数将生成两个（而非一个）空间索引表。其中较小的空间索引表是非页表，会经常在空间分析时被遍历。由于非页表经常被访问，所以将非页表固定在缓冲池中是有好处的。请参见下面的示例。

-- 创建索引

```
CREATE INDEX geod_counties_sidx ON geod_counties(geom)
INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS ('sdo_non_leaf_tbl=TRUE');
```

-- 查找非页索引表名称

```
SELECT sdo_nl_index_table
FROM user_sdo_index_metadata
WHERE sdo_index_name='GEOD_COUNTIES_SIDX';
-----
MDNT_A930$
```

-- 将表固定在内存中

```
ALTER TABLE MDNT_A930$ STORAGE(BUFFER_POOL KEEP);
```

分区的本地空间索引

推荐使用以下方法在分区表上并行创建本地分区空间索引。使用此方法时，如果任何一个 ALTER INDEX 命令失败，您无需重新构建任何已完成的本地分区索引。

- 使用 UNUSABLE 关键字创建 LOCAL 空间索引。该操作很快，并且只创建与索引相关的元数据：

```
CREATE INDEX sp_idx ON my_table (location)
INDEXTYPE IS mdsys.spatial_index PARAMETERS
('tablesapce=tb_name
work_tablespace=work_tb_name')
LOCAL UNSUABLE;
```

- 构建单个脚本。假设有 100 个分区和 10 个处理器，您可以构建 10 个脚本，其中每个脚本有 10 个 ALTER INDEX REBUILD 语句……但不使用 PARALLEL 参数。例如：
 - ALTER INDEX sp_idx REBUILD PARTITION ip1;
 - ALTER INDEX sp_idx REBUILD PARTITION ip2;
 - 等等
- 同时运行这 10 个脚本。每个处理器将处理单个分区的本地索引，而所有处理器将同时处理各自的一组 ALTER INDEX 语句。

注意：不再推荐使用 PARALLEL 关键字在分区表上并行创建空间索引，原因是创建一个分区的 LOCAL 空间索引时，如果任何分区索引失败（例如，表空间已满或其它原因），则必须重新从头开始（无法在停止处继续）。

空间查询

Oracle Locator 和 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。

操作符有：

- SDO_FILTER (search_column, window)
- SDO_RELATE (search_column, window, 'operator specific parameters')
- SDO_ANYINTERACT (search_column, window)
- SDO_INSIDE (search_column, window)

- SDO_WITHIN_DISTANCE (search_column, window, 'operator specific parameters')
- SDO_NN (search_column, window, 'operator specific parameters')

请参见相应文档了解更多操作符。

函数有：

- SDO_GEOM.SDO_AREA
- SDO_GEOM.SDO_LENGTH
- SDO_CS.TRANSFORM
- SDO_LRS.PROJECT_PT

请参见相应文档了解更多函数。

请注意，所有的空间操作符具有类似的参数签名：

- 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（如果存在的话）是包含该操作符特有参数列表的字符串。

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 Oracle Locator 特有的，但正如它们可以改善不包含空间操作符的 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，当查询窗口（空间操作符的第二个参数）来自于表时，请始终指定 /*+ ordered */ 提示。例如，以下查询查找在 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，被传递给操作符的第二个参数的水井位置（水井 ID 1 和 ID 2）来自于表 well_table。这是一个典型的示例，在此应指定 /*+ ordered */ 提示。

当您指定了 ordered 提示时，请以驱动表的顺序在 FROM 子句中列出各表。提供数据给空间操作符第二个参数的表在 FROM 子句中应排在包含被搜索

列的表之前。在这个示例中，`well_table` 在 `FROM` 子句中位于 `chemical_plants` 之前。

空间操作符通过排除与其第一个参数（被搜索列）关联的表中的行来缩小备选结果的范围。当通过空间操作符可缩小查询结果的范围时，如果 `WHERE` 子句中含为空间操作符第一个参数提供数据的表的其它具有索引的列时，使用 `no_index` 提示可能会有所帮助。这在这些具有索引的列不是很具选择性时尤为如此。

例如，假定我们修改了化工厂查询，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` 列上有一个位图索引，该索引在查询中也不会是非常有选择性的。在 `processes_chromium` 索引上提供一个 `no_index` 提示可以帮助 Oracle 优化程序避免同时使用有选择性的空间索引和无选择性的非空间索引。

```
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 推荐收集有关表和索引的统计信息以使用基于成本的查询优化。这可以通过以下过程实现：

- `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。

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，那么打开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，打开详细的街道就实际得多了。

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像素点上，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综合图层（详细多边形的综合版本）。当缩小得非常小时显示综合图层，当放大到非常大时显示详细图层。

缩放控制和综合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综合图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像素点上显示。

使用 **ESRI ArcGIS** 的最佳实践

ESRI 对 Oracle 数据库的支持已超过 12 年，共同客户数以千计。ESRI 从 1995 年开始就成为每个 Oracle 数据库版本的测试站点。

Oracle Locator 和 Oracle Spatial 完全支持 ESRI 的地理数据库，其上的 ArcGIS 特性和性能满足实际应用的需要。大量的共同客户群证实了这一点。最近，在地理空间信息技术协会 (GITA) 2008 年的地理空间基础架构解决方案会议上，Idea Integration 的 Kevin S. Larson 发表的论文《空间数据库速度比较》对这方面的性能进行了量化。

目的是确定哪种数据库和数据类型最适合 ESRI ArcGIS。它在 Oracle 数据库 10.2.0.3 对 ESRI ArcGIS 版本 9.2 进行了测试，测试中使用了所有受 ESRI SDE 支持的 Oracle 数据类型。该论文研究了热缓存和冷缓存的影响，对点、线和多边形数据的良好表现，以及在不同地理区域和数据比例中出现的不同密度的空间数据。执行了两组查询：1) 仅与几何结构数据相关的单一空间数据查询以及 2) 与几何结构数据和关系数据都相关的业务查询。

研究发现，Oracle Locator 及其原生的 Oracle SDO_GEOMETRY 数据类型为 Oracle 数据库上的 ESRI ArcGIS 提供了最一致和最具可伸缩性的性能。它大幅改善了较大数据集（较小数据比例）的性能，并在较低的数据密度下提供近乎相同的性能。SDO_GEOMETRY 的性能优势在单一空间数据查询中更为显著。这些发现和众多客户的报告完全一致。

下面是根据客户使用经验总结出来的最佳实践，特别用于性能调优使用 Oracle Locator 的 ESRI ArcGIS。

DBTUNE 文件

建议将以下属性添加到与 SDO_GEOMETRY 特性类相关的 DBTUNE 关键字中。下面提供了示例。为 SDO_GEOMETRY 特性类设置适当的属性值。

如果数据在每一点都有值，例如 (x,y) 和一个测量值，请确保将 SDO_LB_3、SDO_UB_3 和 SDO_TOLERANCE_3 属性添加到 DBTUNE 关键字中。如果数据在每点有 4 个数字，例如，(x,y,z) 和一个测量值，请确保将 SDO_LB_4、SDO_UB_4 和 SDO_TOLERANCE_4 属性添加到 DBTUNE 关键字中。

关于 3D 数据的重要注意事项：

- 如果数据是 3D 格式的，请检查几何结构查看 Z 坐标值是否总是 0。这有时是公用程序将 CAD 数据载入 SDO_GEOMETRY 中造成的。如果您发现 Z 坐标总是 0，则建议从几何结构中移除所有的 Z 坐标。如果未移除 Z 坐标，将有 1/3 不需要的几何结构数据通过网络传输。换言之，传输 Z 坐标耗费时间，即使它的值总是 0 也是如此。
- 将 CAD 数据载入 SDO_GEOMETRY 的公用程序通常有一个标志，可以将其设置为以 2D 格式上传数据。

ESRI 客户端工具为 SDO_GEOMETRY 特性类创建空间索引时，是在 SDO_INDEX_SHAPE 中指定 SDO_GEOMETRY CREATE INDEX 参数。

```
GEOMETRY_STORAGE "SDO_GEOMETRY"  
SDO_DIMNAME_1    "x"  
SDO_DIMNAME_2    "y"  
SDO_LB_1         -180.000000  
SDO_LB_2         -90.000000  
SDO_SRID         8307  
SDO_TOLERANCE_1  0.05  
SDO_TOLERANCE_2  0.05  
SDO_UB_1         180.000000  
SDO_UB_2         90.000000  
SDO_VERIFY       "FALSE"  
SDO_INDEX_SHAPE  "tablespace=value  
WORK_TABLESPACE=value etc.."
```

DBTUNE 参数存储于数据库。要提取当前的 DBTUNE 文件，可以使用下面的命令。它将一个名为 dbtune_orig.out 的文件放在 SDEHOME\etc\dbtune.orig 中。

```
sdedbtune -o export -f dbtune_orig.out -u sde -p  
sde_password
```

```
copy dbtune_orig.out dbtune.out
```

将前面讨论的 SDO_GEOMETRY 专用的 DBTUNE 参数添加到 dbtune.out 中。然后将新的 dbtune.out 文件加载到数据库中，如下所示：

```
sdedbtune -o import -f dbtune.out -u sde -p sde
```

使用 ESRI 客户端工具创建空间索引

请参见索引创建的最佳实践。ESRI 客户端工具加载数据时，通常可以选择是否创建空间索引。如果您选择让 ESRI 客户端工具创建空间索引，确保合理设置了在本文档 DBTUNE 部分讨论的 DBTUNE 属性，尤其是与空间索引创建有关的属性。

ArcSDE 的主键

ArcSDE 要求每个注册的表在一个声明为 NUMBER(38) 类型的数字列上都有一个唯一的索引。

非 ESRI 客户端工具创建的空间层的注册

以下是使用 ArcSDE 数据字典表注册非 ESRI 客户端工具创建的空间层的点、线和多边形示例。

- 点示例：

```
sdelayer -o register -l bts,shape -e p -C  
oid_,USER -i esri_sde -u oracle_username -p  
oracle_password
```

- 线示例：

```
sdelayer -o register -l road_geo,shape -e sl+ -  
C objectid,USER -i esri_sde -u oracle_username -  
p oracle_password
```

- 多边形示例：

```
sdelayer -o register -l buildings,shape -e a+ -  
C objectid,USER -i esri_sde -u oracle_username -  
p oracle_password
```

从 ArcSDE 取消层的注册

如果 ESRI 客户端工具创建了层，则 `sdelayer -o delete` 会取消注册并删除表。

如果 ESRI 客户端工具没有创建表，则 `sdelayer -o delete` 会取消层的注册并**移除** `USER_SDO_GEOM_METADATA` 中的条目，但不会删除表。**要重新注册空间层，则必须重新填充 `USER_SDO_GEOM_METADATA`。**

下面是一个取消注册构建层的示例：

```
sdelayer -o delete -l buildings,shape -i  
esri_sde -u oracle_username -p oracle_password
```

使用 ESRI 客户端工具时的性能：

如果您在使用 ESRI 客户端工具时怀疑出现性能问题，请首先参阅索引创建的最佳实践。另一个可以考虑的方法是，尝试使用其它可视化工具呈现，以查看不同工具的表现是否一致。如果不同工具间存在性能差异，请查看本文档中的最佳实践以确保您的 ESRI/SDO_GEOMETRY 环境是最佳的。

Oracle 数据库 11g 显著增强了有空间索引的 SDO_GEOMETRY 列上的编辑性能。这可能是另一个值得测试的领域。

甲骨文（中国）软件系统有限公司

北京总部

地址：北京市朝阳区建国门外大街1号，国贸大厦2座2208室
邮编：100004
电话：(86.10) 6535-6688
传真：(86.10) 6505-7505

北京上地6号办公室

地址：北京市海淀区上地信息产业基地，上地西路8号，
上地六号大厦D座702室
邮编：100085
电话：(86.10) 8278-7300
传真：(86.10) 8278-7373

上海分公司

地址：上海市卢湾区湖滨路222号，企业天地商业中心1号楼16层
邮编：200021
电话：(86.21) 2302-3000
传真：(86.21) 6340-6055

广州分公司

地址：广州市天河区北路233号，中信广场53楼5301&5308室
邮编：510613
电话：(86.20) 8513-2000
传真：(86.20) 3877-1026

成都分公司

地址：成都市人民南路二段18号，四川川信大厦20层A&D座
邮编：610016
电话：(86.28) 8619-7200
传真：(86.28) 8619-9573

大连分公司

地址：大连软件园东路23号，大连软件园国际信息中心2号楼
五层502号A区
邮编：116023
电话：(86.411) 8465-6000
传真：(86.411) 8465-6499

济南分公司

地址：济南市泺源大街150号，中信广场11层1113单元
邮编：250011
电话：(86.531) 8518-1122
传真：(86.531) 8518-1133

甲骨文软件研究开发中心（北京）有限公司

地址：北京市海淀区中关村软件园孵化器2号楼A座一层
邮编：100094
电话：(86.10) 8278-6000
传真：(86.10) 8282-6455

甲骨文研究开发中心（深圳）有限公司

地址：深圳市南山区高新南一道飞亚达大厦16层
邮编：518057
电话：(86.755) 8396-5000
传真：(86.755) 8601-3837

沈阳分公司

地址：沈阳市沈河区青年大街219号，华新国际大厦17层D单元
邮编：110016
电话：(86.24) 2396 1175
传真：(86.24) 2396 1033

南京分公司

地址：南京市玄武区洪武北路55号，置地广场19层1911室
邮编：210028
电话：(86.25) 8476-5228
传真：(86.25) 8476-5226

杭州分公司

地址：杭州市西湖区杭大路15号，嘉华国际商务中心702室
邮编：310007
电话：(86.571) 8717-5300
传真：(86.571) 8717-5299

西安分公司

地址：西安市高新区科技二路72号，零壹广场主楼1401室
邮编：710075
电话：(86.29) 8833-9800
传真：(86.29) 8833-9829

福州分公司

地址：福州市五四路158号，环球广场1601室
邮编：350003
电话：(86.591) 8801-0338
传真：(86.591) 8801-0330

重庆分公司

地址：重庆市渝中区邹容路68号，大都会商厦1611室
邮编：400010
电话：(86.23) 6370-8898
传真：(86.23) 6370-8700

深圳分公司

地址：深圳市南山区高新南一道飞亚达大厦16层
邮编：518057
电话：(86.755) 8396-5000
传真：(86.755) 8601-3837

甲骨文亚洲研发中心（上海）

地址：上海市杨浦区淞沪路290号，创智天地10号楼512-516单元
邮编：200433
电话：86-21-6095 2500
传真：86-21-6095 2555



Oracle Locator 和 Oracle Spatial 11g 最佳实践
2008 年 8 月

公司网址: <http://www.oracle.com> (英文)
中文网址: <http://www.oracle.com/cn> (简体中文)
销售中心: 800-810-0161
售后服务热线: 800-810-0366
培训服务热线: 800-810-9931

欢迎访问:
<http://www.oracle.com> (英文)
<http://www.oracle.com/cn> (简体中文)

版权 ©2008 归 Oracle 公司所有。未经允许, 不得以任何形式和手段复制和使用。

本文的宗旨只是提供相关信息, 其内容如有变动, 恕不另行通知。Oracle 公司对本文内容的准确性不提供任何保证, 也不做任何口头或法律形式的其他保证或条件, 包括关于适用性或符合特定用途的所有默示保证和条件。本公司特别声明对本文档不承担任何义务, 而且本文档也不能构成任何直接或间接的合同责任。未经 Oracle 公司事先书面许可, 严禁将此文档为了任何目的, 以任何形式或手段(无论是电子的还是机械的)进行复制或传播。

Oracle 是 Oracle 公司和/或其分公司的注册商标。其他名字均可能是各相应公司的商标。