



管理：空间数据

---

# SAP Sybase IQ 16.0 SP03

文档 ID: DC02012-01-1603-01

最后修订日期: 2013 年 12 月

© 2013 SAP 股份公司或其关联公司版权所有, 保留所有权利。

未经 SAP 股份公司明确许可, 不得以任何形式或为任何目的复制或传播本文的任何内容。本文包含的信息如有更改, 恕不另行事先通知。

由 SAP 股份公司及其分销商营销的部分软件产品包含其它软件供应商的专有软件组件。各国的产品规格可能不同。

上述资料由 SAP 股份公司及其关联公司 (统称“SAP 集团”) 提供, 仅供参考, 不构成任何形式的陈述或保证, 其中如若存在任何错误或疏漏, SAP 集团概不负责。与 SAP 集团产品和服务相关的保证仅限于该等产品和服务随附的保证声明 (若有) 中明确提出之保证。本文中的任何信息均不构成额外保证。

SAP 和本文提及的其它 SAP 产品和服务及其各自标识均为 SAP 股份公司在德国和其它国家的商标或注册商标。

如欲了解更多商标信息和声明, 请访问: <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>。

# 目录

约束和限制 .....	1
空间数据 .....	3
空间参照系 (SRS) 和空间参照标识符 (SRID) .....	4
测量单位 .....	6
安装附加的预定义测量单位 .....	6
SAP Sybase IQ 空间数据支持 .....	7
支持的空间数据类型及其层次 .....	7
对空间标准的遵从性 .....	10
有关支持和遵从性的特殊注意事项 .....	10
所支持的空间数据导入与导出格式 .....	11
对 ESRI shapefile 的支持 .....	12
有关空间主题的推荐阅读材料 .....	13
创建空间列 (SQL) .....	13
空间列的索引 .....	14
空间数据类型语法 .....	14
如何创建几何 .....	17
以图像形式查看空间数据 (Interactive SQL) .....	17
以图像形式查看空间数据 (空间查看器) .....	18
从标准文本 (WKT) 文件装载空间数据 .....	19
创建或管理空间参照系 .....	20
创建或管理空间测量单位 .....	21
高级空间主题 .....	23
平面地球表示和球形地球表示的工作原理 .....	23
对齐网格和公差如何影响空间计算 .....	24
内插值如何影响空间计算 .....	28
多边形环方向的工作原理 .....	29
几何内部、几何外部和几何边界的工作原理 .....	30
空间比较的工作原理 .....	31
空间关系的工作原理 .....	32
空间维度的工作原理 .....	35
教程：试用空间要素 .....	37

第 1 课: 安装附加测量单位和空间参照系 .....	37
第 2 课: 下载 ESRI shapefile 数据 .....	38
第 3 课: 装载 ESRI shapefile 数据 .....	38
第 4 课: 查询空间数据 .....	41
第 5 课: 将空间数据输出为 SVG .....	43
第 6 课: 投影空间数据 .....	45
访问和操作空间数据 .....	49
附录 - SQL 语句 .....	51
CREATE SPATIAL REFERENCE SYSTEM 语句 .....	51
CREATE SPATIAL UNIT OF MEASURE 语句 .....	57
DROP SPATIAL UNIT OF MEASURE 语句 .....	59
DROP SPATIAL REFERENCE SYSTEM 语句 .....	60
ALTER SPATIAL REFERENCE SYSTEM 语句 .....	60
ALTER TABLE 语句 .....	66
索引 .....	81

# 约束和限制

在使用 SAP® Sybase® IQ 中的空间数据、空间参照系和空间测量单位之前，请先熟悉 3D 方法在性能和参照完整性方面的约束和影响。

## 2006 ISO 标准

虽然您可以存储 Z 和 M 维度，但不支持 3D 方法。2006 ISO 标准仅支持 2D 空间方法（X 和 Y 维度）。

### 空间数据必须存储在 IQ 目录存储表中

空间数据、空间参照系和空间测量单位只能在目录存储库中使用。IQ 主存储库无法解释或存储空间数据。可以在目录存储中查询空间数据以及与 IQ 主存储表进行连接，但该连接必须在非空间列中。

例如，假设您想将 ST\_Point 与存储在 IQ 主存储表中的每位客户相关联。由于 IQ 主存储表无法存储 ST\_Point，您必须创建一个单独的 IQ 存储表来保存点：

IQ 主存储表	Customer( CustID, CustName, ... )
IQ 目录存储表	CustPoints( CustID, Point )

假设有一个 ST\_Polygon 为 P，您想通过查询来找出 P 中的所有客户。假设 P 是一个连接变量，该变量已通过常量或先前的 IQ 目录存储表查询进行填充。

```
Select C.*
From Customer C, CustPoints CP
Where C.CustID = CP.CustID
And CP.Point.ST_Within( P ) = 1
```

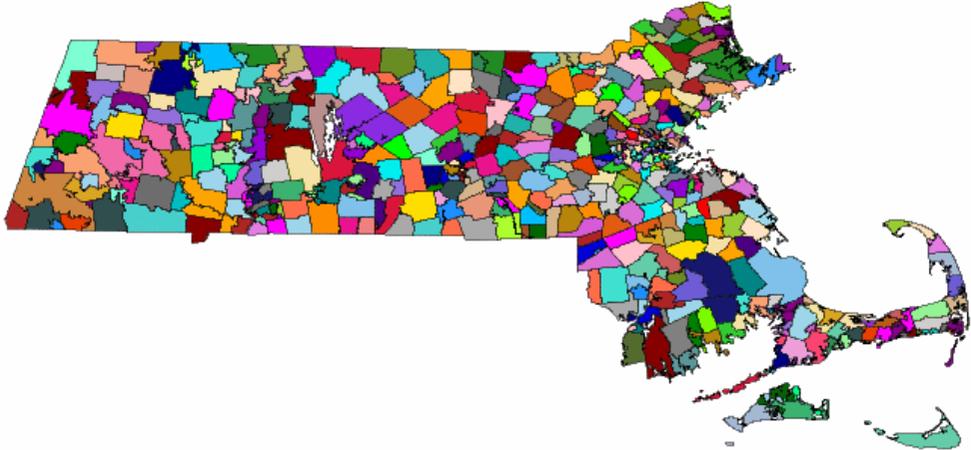
由于该查询会使 Customer 表从 IQ 主存储库流入 IQ 目录存储，因此性能会受到影响。需要考虑 CIS 函数补偿性能方面的一些事宜。

参照完整性约束未在 IQ 主存储库/IQ 目录存储桥间进行维护。请确保一个表中的行插入/删除反映到连接表中。例如，每个 CustPoints 行必须对应于一个 Customer 行。如果删除 Customer 行，必须删除相应的 CustPoints 行。



# 空间数据

**空间数据**是用于描述所定义空间中对象的位置、形状和方向的数据。SAP Sybase IQ 中的空间数据表示为点、曲线（线串和圆弧线串）和多边形形式的 2D 几何。例如，下图显示了马萨诸塞州，它是表示各邮政编码区域的多边形的联合。



对空间数据执行的两个常见操作是计算几何之间的距离和确定多个对象之间的联合或相交。使用谓词（如相交、包含和交叉）执行这些计算。

空间数据文档假定您对空间参照系以及想要使用的空间数据已有所了解。

---

**注意：**支持 32 位 Windows 和 32 位 Linux 的空间数据需要一个支持 SSE2 指令的 CPU。使用 Intel Pentium 4 或更高版本（2001 年发布）以及 AMD Opteron 或更高版本（2003 年发布）可获得此项支持。

---

## 有关如何使用空间数据的示例

SAP Sybase IQ 中提供的空间数据支持可帮助应用程序开发人员将空间信息与其数据相关联。例如，表示公司的表可以将公司位置存储为点，或将公司的送货区域存储为多边形。在 SQL 中上述假设可表示为：

```
CREATE TABLE Locations(  
    ID INT,  
    ManagerName CHAR(16),  
    StoreName CHAR(16),  
    Address ST_Point,  
    DeliveryArea ST_Polygon )
```

在本例中，空间数据类型 **ST\_Point** 表示单个点，**ST\_Polygon** 表示任意多边形。利用此模式，应用程序可以在地图上显示所有公司位置，或者使用以下形式的查询来确定公司是否向特定地址送货：

```
CREATE VARIABLE @pt ST_Point;
SET @pt = ST_Geometry::ST_GeomFromText( 'POINT(1 1)' );

SELECT * FROM Locations
WHERE DeliveryArea.ST_Contains( @pt ) = 1
```

SAP Sybase IQ 提供空间数据的存储和数据管理功能，从而能够存储诸如地理位置、路线信息和形状数据之类的信息。

这些信息片段可以作为点、各种形状的多边形和线，存储在使用相应**空间数据类型**（如 `ST_Point` 和 `ST_Polygon`）定义的列中。使用方法和构造程序来访问和操作空间数据。SAP Sybase IQ 还提供了一组旨在与其它产品兼容的 **SQL 空间函数**。

## 空间参照系 (SRS) 和空间参照标识符 (SRID)

在空间数据库的上下文中，用来描述几何的定义空间被称为**空间参照系 (Spatial Reference System, 简称 SRS)**。空间参照系至少定义以下内容：

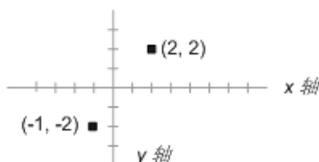
- 基础坐标系的测量单位（度、米等）
- 最大坐标值和最小坐标值（也称为边界）
- 缺省线性测量单位
- 数据是平面数据还是椭球体数据
- 用于将数据转换为其它 SRS 的投影信息

每个空间参照系均有一个标识符，称为**空间参照标识符 (Spatial Reference Identifier, 简称 SRID)**。当 SAP Sybase IQ 执行操作时（如确定一个几何是否与另一个几何相接触），将使用 **SRID** 查找空间参照系定义，以确保正确执行对空间参照系的计算。在 SAP Sybase IQ 数据库中，每个 **SRID** 必须唯一。

缺省情况下，SAP Sybase IQ 向新数据库添加以下空间参照系：

- **缺省 - SRID 0** – 这是构造几何时使用的缺省空间参照系，并且 **SRID** 不在 **SQL** 中指定，也不存在于正在装载的值中。

缺省空间参照系是在二维平面上使用数据的笛卡尔空间参照系。平面上的任一点均可使用一对 **x, y** 坐标来定义，其中 **x** 和 **y** 的边界为 **-1,000,000** 到 **1,000,000**。使用垂直坐标轴测量距离。为此缺省空间参照系指定的 **SRID** 为 **0**。



笛卡尔是平面类型的空间参照系。

- **WGS 84 - SRID 4326** – WGS 84 标准提供地球的球体参照面。这是全球定位系统 (Global Positioning System, 简称 GPS) 使用的空间参照系。WGS 84 的坐标原点

是地球的质心，精度可达到  $\pm 1$  米。WGS 表示世界坐标系 (World Geodetic System)。

WGS 84 坐标单位是度，其中，第一个坐标是经度，范围是 -180 到 180；第二个坐标是纬度，范围是 -90 到 90。

WGS 84 的缺省测量单位是米，是球形地球类型的空间参照系。

- **WGS 84 (平面) - SRID 1000004326** - WGS 84 (平面) 与 WGS 84 类似，不同之处是此参照系使用等矩形投影，此投影方法会使长度、面积和其它计算项失真。例如，在 SRID 4326 和 SRID 1000004326 中，在赤道上，1 度经度大约为 111 km。在 SRID 4326 中，在 80 度北纬线上，1 度经度大约为 19 km；但是，SRID 1000004326 却将所有纬度上的 1 度经度均视为大约 111 km。SRID 1000004326 中的长度失真量相当大—失真系数达到了 10 或更大—失真系数会根据几何相对于赤道的位置而变化。因此，SRID 1000004326 不能用于计算距离和面积。此参照系只能用于关系谓词，如 ST\_Contains、ST\_Touches、ST\_Covers 等。

WGS 84 (平面) 的缺省测量单位是度，是平面地球类型的空间参照系。

- **sa\_planar\_unbounded - SRID 2,147,483,646** - 仅供内部使用。
- **sa\_octahedral\_gnomonic - SRID 2,147,483,647** - 仅供内部使用。

由于您可以随意定义空间参照系并可以指定任意 SRID 编号，因此在数据库之间移动数据时或者将数据转换为其它空间参照系时，空间参照系定义 (投影、坐标系等) 必须随数据一起移动。例如，将空间数据卸载到 WKT 时，在文件的开头会包括空间参照系的定义。

#### *使用 sa\_install\_feature 系统过程安装附加空间参照系*

SAP Sybase IQ 还提供数千个预定义的 SRS 供用户使用。不过，缺省情况下，创建新数据库时这些 SRS 不会安装到数据库中。但可以使用 sa\_install\_feature 系统过程进行添加。

可以在 [spatialreference.org](http://spatialreference.org) 和 [www.epsg-registry.org/](http://www.epsg-registry.org/) 中找到这些附加空间参照系的说明。

#### *确定当前在数据库中的空间参照系的列表*

空间参照系信息存储在 ISYSSPATIALREFERENCESYSTEM 系统表中。SRS 的 SRID 用作此表中的主键值。数据库服务器使用 SRID 值查找空间参照系的配置信息，以便可以将其它抽象的空间坐标解释为地球上的实际位置。

通过查询 ST\_SPATIAL\_REFERENCE\_SYSTEMS 统一视图可以查找空间参照系列表。此视图中的每一行都定义一个空间参照系。

#### *与流行制图应用程序的兼容性*

Google 地球、Bing 地图和 ArcGIS Online 等流行 Web 制图和可视化应用程序使用空间参照系和基于地球球面模型的墨卡托投影。此球面模型忽略地球两极处的扁率，因此会导致位置误差高达 800m，比例误差高达 0.7%，但它也提高了应用程序进行投影的效率。

过去，商业应用程序会将 SRID 900913 分配给此空间参照系。然而，EPSG 已将此投影作为 SRID 3857 发布。为了与需要 900913 的应用程序兼容，您可以执行以下操作：

1. 使用 `sa_install_feature` 系统过程来安装 SAP Sybase IQ 提供的所有空间参照系（包括 SRID 3857）。
2. 执行 `[dbunload -n]`，获取 3857 SRID 定义（SAP Sybase IQ 中未提供 `dbunload` 实用程序）。

## 测量单位

---

地理要素可以使用纬度的度、弧度或其它角度测量单位进行测量。每个空间参照系都必须明确规定测量地理坐标所使用的单位名称，并且必须包括从指定单位到弧度的转换。

如果使用投影坐标系，则各坐标值表示沿地球表面测量的到某一点的线性距离。坐标值可以使用米、英尺、英里或码为单位来测量。投影坐标系必须明确规定用于表示坐标值的线性测量单位。

以下测量单位会自动安装到任何新 SAP Sybase IQ 数据库中：

- **米 (meter)** - 线性测量单位。也称作国际米。SI 标准单位。由 ISO 1000 定义。
- **米 (metre)** - 线性测量单位。`meter` (米) 的别名。SI 标准单位。由 ISO 1000 定义。
- **弧度** - 角度测量单位。SI 标准单位。由 ISO 1000:1992 定义。
- **度** - 角度测量单位 ( $\pi/180.0$  弧度)。
- **平面度** - 线性测量单位。定义为 60 海里。具有平面线解释的地理空间参照系所使用的线性测量单位。

## 安装附加的预定义测量单位

`sa_install_feature` 系统过程可添加新数据库中未缺省安装的附加预定义测量单位。

### 前提条件

无。

### 过程

执行以下语句，安装所有预定义测量单位：

```
CALL sa_install_feature('st_geometry_predefined_uom');
```

所有附加测量单位安装完成。

### 下一步

您可以创建一个使用测量单位的空间参照系。

您可以在 [www.epsg-registry.org/](http://www.epsg-registry.org/) 中找到这些附加测量单位的说明。在 Web 页上，于 [Name] 字段中键入测量单位的名称，从 [Type] 字段选择 [Unit of Measure (UOM)]，然后单击 [Search]。

## SAP Sybase IQ 空间数据支持

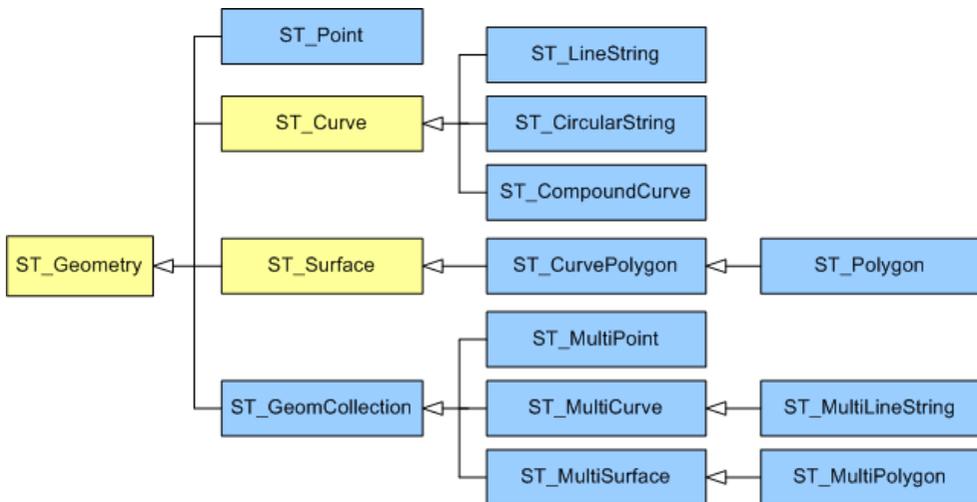
以下几节介绍 SAP Sybase IQ 空间数据支持。

### 支持的空间数据类型及其层次

SAP Sybase IQ 遵守存储和访问地理空间数据的 SQL 多媒体 (SQL/MM) 标准。此标准的一个关键组成部分是使用 `ST_Geometry` 类型层次来定义如何创建地理空间数据。在该层次内，所有数据类型（也称为类或类型）均使用前缀 `ST_`。

当某列被标识为某个特定类型时，则该类型及其子类型的值均可存储在该列中。例如，被标识为 `ST_GeomCollection` 的列也可存储 `ST_MultiPoint`、`ST_MultiSurface`、`ST_MultiCurve`、`ST_MultiPolygon` 和 `ST_MultiLineString` 值。

下图说明了 `ST_Geometry` 数据类型及其子类型的层次：



图例： 可实例化  不可实例化 A ← B B 为 A 的子类型

左侧的类型是右侧的子类型（或派生类型）的超类型（或基本类型）。

#### 所支持空间数据类型的描述

SAP Sybase IQ 支持以下空间数据类型：

- **点** - 点用于定义空间中的单个位置。点几何没有长度或面积。点始终有 X 和 Y 坐标。

对于非空点，`ST_Dimension` 返回 0。

在 GIS 数据中，点通常用于表示位置（例如地址）或地理要素（例如山脉）。

- **线串** - 线串是具有长度但没有面积的几何。对于非空线串，`ST_Dimension` 返回 1。可以指定线串是否具有简单特征，以及是否具有闭合特征。**简单**表示线串不穿越自身。**闭合**表示线串的起点和终点为同一点。例如，环就是一种简单的闭合线串。

在 GIS 数据中，线串通常用于表示河流、道路或货运路线。

- **多边形** - 多边形定义空间区域。多边形由一个定义区域外界的外围环和零个或多个定义区域内的孔的内部环构造而成。多边形具有关联的面积，但没有长度。

对于非空多边形，`ST_Dimension` 返回 2。

在 GIS 数据中，多边形通常用于表示地域（县、市、省等）、湖泊和较大的地理要素（如公园）。

- **圆弧线串** - 圆弧线串是一连串连接的圆弧线段；圆弧线串与线串非常相似，只是其两点之间是圆弧。
- **复合曲线** - 复合曲线是一连串连接的圆弧线串或线串。
- **曲线多边形** - 曲线多边形是更普通的多边形，其边界线可能有圆弧线段。
- **几何** - 术语“几何”表示对象（如点、线串和多边形）的全局类型。“几何”类型是所支持的所有空间数据类型的超类型。
- **几何集合** - 几何集合是包含一个或多个几何（例如点、线、多边形等）的集合。
- **多点** - 多点是单个点的集合。

在 GIS 数据中，多点通常用于表示一组位置。

- **多面** - 多面是零个或多个多边形的集合。

在 GIS 数据中，多面常用于表示由多个区域组成的地域（例如由多个岛屿组成的州）或地理要素（如湖系）。

- **多线串** - 多线串是线串的集合。

在 GIS 数据中，多线串通常用于表示地理要素，如河流或公路网。

- **多表面** - 多表面是曲线多边形的集合。

#### *空间数据类型的面向对象属性*

- 子类型（或派生类型）比其超类型（或基本类型）更具体。例如，`ST_LineString` 是 `ST_Curve` 的更具体的类型。
- 子类型继承了所有超类型的所有方法。例如，`ST_Polygon` 值可调用 `ST_Geometry`、`ST_Surface` 和 `ST_CurvePolygon` 超类型的方法。
- 子类型的值可自动转换为其超类型的任何值。例如，`ST_Point` 值可用于需要 `ST_Geometry` 参数之处，如 `[point1.ST_Distance( point2 )]`。
- 列或变量可以存储任何子类型的值。例如，类型为 `ST_Geometry(SRID=4326)` 的列可以存储任意类型的空间数据。
- 可将具有声明类型的列、变量或表达式视为或转换为子类型。例如，可使用 `TREAT` 表达式将名为 `geom` 的 `ST_Geometry` 列中的 `ST_Polygon` 值更改为具有声明类型

`ST_Surface`，以便使您能够使用 `[TREAT( geom AS ST_Surface ).ST_Area()]` 调用其中的 `ST_Area` 方法。

### 所支持的空间谓词

谓词是用逻辑运算符 `AND` 和 `OR` 连接在一起的条件表达式，可在 `WHERE`、`HAVING` 或 `ON` 子句中（或在 `IF` 或 `CASE` 表达式中，或是在 `CHECK` 约束中）组成条件集合。在 `SQL` 中，谓词的值可以是 `TRUE` 或 `FALSE`。在许多上下文中，求值结果为 `UNKNOWN` 的谓词将解释为 `FALSE`。

空间谓词被实现为可返回 0 或 1 的成员函数。为测试空间谓词，查询应使用 `=` 或 `<>` 运算符将函数的结果与 1 或 0 进行比较。例如：

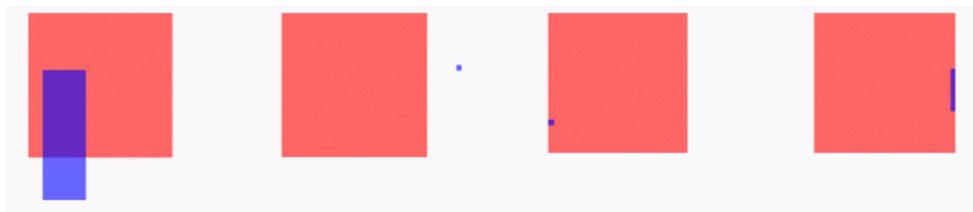
```
SELECT * FROM SpatialShapes WHERE geometry.ST_IsEmpty() = 0;
```

当查询空间数据以回答如下问题时使用谓词：两个或多个几何的接近程度如何？它们是否相交或重叠？一个几何是否包含在另一个几何中？举例来说，如果用户是一家货运公司，则用户可以使用谓词来判定客户是否在特定送货区域内。

### 空间谓词的直观性

有时谓词的结果并不直观，因此您应该对某些特殊情况进行测试，以确保得到您所期望的结果。例如，要使一个几何包含另一个几何 (`a.ST_Contains(b)=1`)，或使一个几何位于另一个几何内部 (`b.ST_Within(a)=1`)，则 `a` 的内部和 `b` 的内部必须相交，且 `b` 的任何部分都不得与 `a` 的外部相交。然而，有时，您会希望一个几何被视为包含另一个几何或位于另一个几何内，但测试结果却并非如此。

对于 `a.ST_Contains(b)` 和 `b.ST_Within(a)`，下例均返回 0 (`a` 为红色)：



情况一和情况二显而易见；紫色几何并不完全在红色正方形内。而情况三和情况四则不太明显。在这两种情况中，紫色几何仅处于红色正方形的边界上。即使紫色几何看上去在红色正方形内，`ST_Contains` 也认为紫色几何不在红色正方形内。

`ST_Covers` 和 `ST_CoveredBy` 是与 `ST_Contains` 和 `ST_Within` 类似的谓词。其不同之处在于：`ST_Covers` 和 `ST_CoveredBy` 不要求两个几何的内部相交。此外，`ST_Covers` 和 `ST_CoveredBy` 产生的结果通常比 `ST_Contains` 和 `ST_Within` 产生的结果更直观。

如果谓词测试针对各个情况返回的结果不同于所需结果，请考虑使用 `ST_Relate` 方法来指定要测试的确切关系。

## 对空间标准的遵从性

SAP Sybase IQ 空间遵从下列标准：

- **国际标准化组织 (ISO)** – SAP Sybase IQ 几何符合用于定义空间用户类型、例程和模式以及用于处理空间数据的 ISO 标准。SAP Sybase IQ 符合国际标准 ISO/IEC 13249-3:2006 的特定建议。请参见 [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=38651](http://www.iso.org/iso/catalogue_detail.htm?csnumber=38651)。
- **开放地理空间协会 (OGC) 几何模型** – SAP Sybase IQ 几何符合地理信息 OGC OpenGIS 实现规范 - 简单功能访问 - 第二部分：SQL 选项 1.2.0 版 (OGC 06-104r3)。请参见 <http://www.opengeospatial.org/standards/sfs>。

SAP Sybase IQ 使用 OGC 建议的标准来确保空间信息可以在不同的供应商和应用程序之间共享。

为了确保能够兼容 SAP Sybase IQ 空间几何，建议您遵守 OGC 指定的标准。

- **SQL 多媒体 (SQL/MM)** – SAP Sybase IQ 遵守 SQL/MM 标准，并在所有方法和函数名称中使用前缀 ST\_。

SQL/MM 是定义如何使用 SQL 存储、检索和处理空间数据的国际标准。空间数据类型层次（例如 ST\_Geometry）是用于检索空间数据的方法之一。ST\_Geometry 层次包括许多子类型，例如 ST\_Point、ST\_Curve 和 ST\_Polygon。使用 SQL/MM 标准，查询中包括的每个空间值都必须在同一空间参照系中进行定义。

## 有关支持和遵从性的特殊注意事项

本节描述有关 SAP Sybase IQ 支持空间数据方面的特殊注意事项，包括不支持的功能和明显不同于其它数据库产品的行为差异。

- **地理与几何** – 一些供应商通过判断空间对象是**地理**（与球形地球上的对象有关）还是**几何**（平面或平面地球上的对象）来对它们加以区分。在 SAP Sybase IQ 中，所有空间对象均被视为几何，对象的 SRID 用于指示是在球形地球空间参照系还是在平面地球（平面）空间参照系中操作对象。
- **不支持的方法** –

- ST\_Buffer 方法
- ST\_LocateAlong 方法
- ST\_LocateBetween 方法
- ST\_Segmentize 方法
- ST\_Simplify 方法
- ST\_Distance\_Spheroid 方法
- ST\_Length\_Spheroid 方法

## 所支持的空间数据导入与导出格式

下表列出了 SAP Sybase IQ 所支持的用于导入和导出空间数据的数据和文件格式：

日期格式	导入	导出	说明
标准文本 (WKT)	是	是	<p>以 ASCII 文本形式表示的地理数据。此格式作为为地理信息 OpenGIS 实现规范定义的简单功能的一部分由开放地理空间协会 (OGC) 进行维护。请参见 <a href="http://www.opengeospatial.org/standards/sfa">www.opengeospatial.org/standards/sfa</a>。</p> <p>下面的示例显示了如何以 WKT 格式表示点：</p> <pre>'POINT (1 1)'</pre>
标准二进制 (WKB)	是	是	<p>以二进制流形式表示的地理数据。此格式作为为地理信息 OpenGIS 实现规范定义的简单功能的一部分由 OGC 进行维护。请参见 <a href="http://www.opengeospatial.org/standards/sfa">www.opengeospatial.org/standards/sfa</a>。</p> <p>下面的示例显示了如何以 WKB 格式表示点：</p> <pre>'010100000000000000000000F03F000000000000F03F'</pre>
扩展标准文本 (EWKT)	是	是	<p>WKT 格式，但嵌入了 SRID 信息。此格式作为 PostGIS (PostgreSQL 的空间数据库扩展) 的一部分来进行维护。请参见 <a href="http://postgis.refrains.net/">postgis.refrains.net/</a>。</p> <p>下面的示例显示了如何以 EWKT 格式表示点：</p> <pre>'srid=101;POINT (1 1)'</pre>
扩展标准二进制 (EWKB)	是	是	<p>WKB 格式，但嵌入了 SRID 信息。此格式作为 PostGIS (PostgreSQL 的空间数据库扩展) 的一部分来进行维护。请参见 <a href="http://postgis.refrains.net/">postgis.refrains.net/</a>。</p> <p>下面的示例显示了如何以 EWKB 格式表示点：</p> <pre>'01010000020040000000000000000000F03F000000000000F03F'</pre>
地理标记语言 (GML)	否	是	<p>XML 语法曾用于表示地理空间数据。此标准由开放地理空间协会 (OGC) 进行维护，用于通过 Internet 交换地理数据。请参见 <a href="http://www.opengeospatial.org/standards/gml">www.opengeospatial.org/standards/gml</a>。</p> <p>下面的示例显示了如何以 GML 格式表示点：</p> <pre>&lt;gml:Point&gt; &lt;gml:coordinates&gt;1,1&lt;/gml:coordinates&gt; &lt;/gml:Point&gt;</pre>

日期格式	导入	导出	说明
KML	否	是	<p>此 XML 语法的前身为 Google 的 Keyhole 标记语言，用于表示地理数据，包括可视化和导航辅助工具以及为地图和图像添加标注的功能。Google 向 OGC 推荐了此标准。OGC 将其用作开放标准，此标准现在称为 KML。请参见 <a href="http://www.opengeospatial.org/standards/kml">www.opengeospatial.org/standards/kml</a>。</p> <p>下面的示例显示了如何以 KML 格式表示点：</p> <pre>&lt;Point&gt; &lt;coordinates&gt;1,0&lt;/coordinates&gt; &lt;/Point&gt;</pre>
ESRI shapefile	是	否	<p>以 shapefile（用来共同定义形状的几个文件）形式表示空间对象的流行地理空间矢量数据格式。</p>
GeoJSON	否	是	<p>一种文本格式，它使用名称/值对、值的有序列表以及在常用编程语言（例如 C、C++、C#、Java、JavaScript、Perl 和 Python）中使用的类似约定。</p> <p>GeoJSON 是 JSON 标准的一个子集，用于对地理信息进行编码。SAP Sybase IQ 支持 GeoJSON 标准并提供了 ST_AsGeoJSON 方法，用于将 SQL 输出转换为 GeoJSON 格式。</p> <p>下面的示例显示了如何以 GeoJSON 格式表示点：</p> <pre>{"x" : 1, "y" : 1, "spatialReference" : {"wkid" : 4326}}</pre> <p>有关 GeoJSON 说明的详细信息，请参见 <a href="http://geojson.org/geojson-spec.html">geojson.org/geojson-spec.html</a>。</p>
可缩放矢量图形 (SVG) 文件	否	是	<p>基于 XML 的格式，用于表示二维几何。SVG 格式由万维网联盟 (W3C) 进行维护。请参见 <a href="http://www.w3.org/Graphics/SVG/">www.w3.org/Graphics/SVG/</a>。</p> <p>下面的示例显示了如何以 SVG 格式表示点：</p> <pre>&lt;rect width="1" height="1" fill="deepskyblue" stroke="black" stroke-width="1" x="1" y="-1"/&gt;</pre>

## 对 ESRI shapefile 的支持

SAP Sybase IQ 支持 Environmental System Research Institute, Inc. (ESRI) shapefile 格式。ESRI shapefile 用于为数据集中的空间要素存储几何数据和属性信息。

ESRI shapefile 至少包括三种不同的文件：.shp、.shx 和 .dbf。主文件的后缀是 .shp，索引文件的后缀是 .shx，属性列的后缀是 .dbf。所有文件共用同一个基本名，且通常被组合在一个单一的压缩文件中。SAP Sybase IQ 可以读取具有除多面形 (MultiPatch) 之外所有形状类型的 ESRI shapefile。其中包括包含 Z 和 M 数据的形状类型。

ESRI shapefile 中的数据通常包含多个行和列。例如，空间教程加载了包含马萨诸塞州邮政编码区域的 shapefile。对于每个邮政编码区域，该 shapefile 文件中都包含一行

内容，包括区域的多边形信息。该 `shapefile` 还包含每个邮政编码区域的附加属性（列），包括邮政编码名称（例如，字符串“02633”）和其它属性。

将 `shapefile` 装载到表中的最简单方法是使用 Interactive SQL “导入向导” 或 `st_geometry_load_shapefile` 系统过程。这两个工具可创建具有相应列的表，并装载 `shapefile` 的数据。

还可以使用 `LOAD TABLE` 和 `INPUT` 语句装载 `shapefile`，但必须在执行装载操作之前已创建具有相应列的表。

要确定使用 `LOAD TABLE` 或 `INPUT` 语句装载数据时所需的列，可使用 `sa_describe_shapefile` 系统过程。

有关 ESRI `shapefile` 的详细信息，请参见 <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>。

## 有关空间主题的推荐阅读材料

如果希望通过一本优秀的初级读物来了解有关制图和测量地球表面（大地测量学）的不同方法，以及坐标（或空间）参照系的主要概念，请访问 [www.epsg.org/guides/index.html](http://www.epsg.org/guides/index.html) 并选择 Geodetic Awareness。

地理信息 OGC OpenGIS 实现规范 - 简单功能访问：[www.opengeospatial.org/standards/sfs](http://www.opengeospatial.org/standards/sfs)

国际标准 ISO/IEC 13249-3:2006：[www.iso.org/iso/catalogue\\_detail.htm?csnumber=38651](http://www.iso.org/iso/catalogue_detail.htm?csnumber=38651)

可缩放矢量图形 (SVG) 1.1 规范：[www.w3.org/Graphics/SVG/](http://www.w3.org/Graphics/SVG/)

地理标记语言 (GML) 规范：[www.opengeospatial.org/standards/gml](http://www.opengeospatial.org/standards/gml)

KML 规范：[www.opengeospatial.org/standards/kml](http://www.opengeospatial.org/standards/kml)

JavaScript Object Notation (JSON)：[json.org](http://json.org)

GeoJSON 规范：[geojson.org/geojson-spec.html](http://geojson.org/geojson-spec.html)

## 创建空间列 (SQL)

可以通过添加一个支持空间数据的列将空间数据添加到任意表。

### 前提条件

您必须是表的所有者、对表具有 `ALTER` 特权，或者具有 `ALTER ANY TABLE` 或 `ALTER ANY OBJECT` 系统特权。

### 过程

1. 连接到数据库。
2. 执行 `ALTER TABLE` 语句。

一个空间列将添加到现有表中。

### 下一步

可以在列上添加 **SRID** 约束，以便对可存储在空间列中的值加以限制。

## 空间列的索引

---

与为任何其它数据类型创建索引时一样，创建空间索引时可使用 **CREATE INDEX** 语句或“**创建索引向导**”。但是，对空间数据创建索引时，建议您不要在索引中包括多个空间列，并且应该将空间列放在索引定义的最后一行。

另外，要在索引中包括某个空间列，该列必须具有 **SRID** 约束。

对空间数据建立索引可降低评估几何之间关系的花销。例如，假设您正在考虑改变销售区域的边界并想要确定对现有客户的影响。要确定哪些客户在提议的销售区域内，可以使用 **ST\_Within** 方法将表示每个客户地址的点与表示销售区域的多边形进行比较。没有建立索引时，数据库服务器必须对照销售区域多边形测试 **Customer** 表中的每个地址点，以确定是否应该在结果中返回该点。如果 **Customer** 表很大，则操作的花销可能非常大；如果销售区域很小，则效率又非常低。如果索引包括每个客户的地址点，则有助于快速返回结果。如果可以向查询添加谓词从而将销售区域与销售区域所覆盖的州相关联，则使用包括州代码和地址点的索引可以更快速地获得结果。

空间查询 *可能会* 从聚簇索引中受益，但在决定使用聚簇索引之前，需要考虑表的其它应用。应该考虑和测试可能执行的查询类型，以确定使用聚簇索引时性能是否提高。

虽然可以对空间列创建文本索引，但文本索引并不比常规索引更具优势，因此改用常规索引。

---

**注意：** 空间列不能包括在主键、唯一索引或唯一约束中。

---

## 空间数据类型语法

---

SQL/MM 标准根据建立于 ANSI/SQL **CREATE TYPE** 语句基础之上的用户定义扩展类型 (UDT) 定义空间数据支持。虽然 SAP Sybase IQ 不支持用户定义的类型，但仍然实现了 SAP Sybase IQ 空间数据支持，就好像它们受支持一样。

### 实例化 UDT 实例

可通过调用构造函数来实例化用户定义类型的值，如下所示：

```
NEW type-name( argument-list)
```

例如，查询可以包含以下内容以实例化两个 **ST\_Point** 值：

```
SELECT NEW ST_Point(), NEW ST_Point(3,4)
```

SAP Sybase IQ 使用常规重载解决规则对照已定义的构造程序来匹配 *argument-list*。出现以下情况时会返回错误：

- **NEW** 与不是用户定义类型的类型一起使用
- 无法实例化用户定义类型（例如，**ST\_Geometry** 不是可实例化的类型）。
- 没有与提供的参数类型相匹配的重载

#### 使用实例方法

用户定义类型可以定义实例方法。在类型的值上调用实例方法如下：

```
value-expression.method-name( argument-list )
```

例如，以下虚构的示例会选择 **Massdata.CenterPoint** 列的 X 坐标：

```
SELECT CenterPoint.ST_X() FROM Massdata;
```

如果存在名为 **CenterPoint** 的用户 ID，数据库服务器会将 [**CenterPoint.ST\_X()**] 视为**不明确**。这是因为该语句可能表示 "调用用户 **CenterPoint** 拥有的用户定义函数 **ST\_X**"（不是此语句的正确意图），也可能表示 "调用 **Massdata.CenterPoint** 列上的 **ST\_X** 方法"（该语句的正确意思）。数据库服务器解决此类不明确问题的方法是首先执行区分大小写的搜索来查找名为 **CenterPoint** 的用户。如果找到用户，则数据库服务器将继续，就好像正在调用一个用户定义的且由用户 **centerpoint** 所拥有的函数 **ST\_X** 一样。如果未找到用户，则数据库服务器会将该结构视为方法调用并对 **Massdata.CenterPoint** 列调用 **ST\_X** 方法。

出现以下情况时调用实例方法会给出错误：

- *value-expression* 声明的类型不是用户定义的类型
- *value-expression* 声明的类型或其某个超类型中未定义指定方法
- *argument-list* 与为指定方法定义的某个重载不匹配。

#### 使用静态方法

除了实例方法，ANSI/SQL 标准还可以将用户定义类型与静态方法关联。使用以下语法调用静态方法：

```
type-name::method-name( argument-list )
```

例如，以下语句通过分析文本将 **ST\_Point** 实例化：

```
SELECT ST_Geometry::ST_GeomFromText('POINT( 5 6 )')
```

出现以下情况时调用静态方法会给出错误：

- *value-expression* 声明的类型不是用户定义的类型
- *value-expression* 声明的类型或其某个超类型中未定义指定方法
- *argument-list* 与为指定方法定义的某个重载不匹配

#### 使用静态集合方法 (SAP Sybase IQ 扩展)

作为 ANSI/SQL 的扩展，SAP Sybase IQ 支持实现用户定义集合的静态方法。例如：

```
SELECT ST_Geometry::ST_AsSVGAggr(T.geo) FROM table T
```

静态方法的所有重载都必须是集合的或者任何重载都不是集合的。

出现以下情况时调用静态集合方法会给出错误：

- 静态方法调用给出错误
- 内置集合函数给出错误
- 指定了 **WINDOW** 子句

#### 使用类型谓词

ANSI/SQL 标准定义的类型谓词允许语句检查值的具体类型（在其它语言中也称为对象类型）。语法如下：

```
value IS [ NOT ] OF ( [ ONLY ] type-name, ...)
```

如果 *value* 为 **NULL**，则谓词返回 **UNKNOWN**。否则，将 *value* 的具体类型与 *type-name* 列表中的每个元素进行比较。如果指定 **ONLY**，则具体类型正好是指定类型时存在匹配项。否则，具体类型是指定类型或任何派生类型（子类型）时存在匹配项。

如果 *value* 的具体类型与列表中的一个元素相匹配，则返回 **TRUE**，否则返回 **FALSE**。

以下示例返回 **Shape** 列值的具体类型为 **ST\_Curve** 或其子类型（**ST\_LineString**、**ST\_CircularString** 或 **ST\_CompoundCurve**）之一的所有行：

```
SELECT * FROM SpatialShapes WHERE Shape IS OF ( ST_Curve );
```

#### 使用子类型的 **TREAT** 表达式

ANSI/SQL 标准定义的子类型处理表达式允许将表达式的声明类型从超类型高效转换为子类型。当您知道表达式的具体类型（在其它语言中也称为对象类型）是指定子类型或指定子类型的子类型时，可使用此表达式。使用 **CAST** 函数时会更有效，因为 **CAST** 函数制作值的副本，而 **TREAT** 不制作副本。语法如下：

```
TREAT ( value-expression AS target-subtype )
```

如果没有出现错误，则结果是声明的类型为 *target-subtype* 的 *value-expression*。

出现以下情况时子类型处理表达式会给出错误：

- *value-expression* 不是用户定义的类型
- 如果 *target-subtype* 不是 *value-expression* 的声明类型的子类型
- 如果 *value-expression* 的动态类型不是 *target-subtype* 的子类型

以下示例将 **ST\_Geometry** **Shape** 列的声明类型有效地更改为 **ST\_Curve** 子类型，以便能够调用 **ST\_Curve** 类型的 **ST\_Length** 方法：

```
SELECT ShapeID, TREAT ( Shape AS ST_Curve ).ST_Length() FROM SpatialShapes WHERE Shape IS OF ( ST_Curve );
```

## 如何创建几何

可以使用以下几种方法在数据库中创建几何：

- **从标准文本 (WKT) 或标准二进制 (WKB) 格式装载** - 可以装载或插入 WKT 或 WKB 格式的数据。这些格式是由 OGC 定义的，所有空间数据库服务商都支持这些格式。SAP Sybase IQ 可自动将这些格式转换成几何类型。
- **从 ESRI shapefile 装载** - 可将数据从 ESRI shapefile 装载到新的或现有的表中。有几种方法可执行此操作。
- **使用 SELECT...FROM OPENSTRING 语句** - 可对包含空间数据的文件执行 SELECT... FROM OPENSTRING 语句。例如：

```
INSERT INTO world_cities( country, city, point )
  SELECT country, city, NEW ST_Point( longitude, latitude, 4326 )
  FROM OPENSTRING( FILE 'capitalcities.csv' )
  WITH (
    country CHAR(100),
    city CHAR(100),
    latitude DOUBLE,
    longitude DOUBLE )
```

- **通过组合经度值和纬度值创建坐标点** - 可通过组合经度值和纬度值来创建空间数据类型 ST\_Point 的坐标。例如，如果您所拥有的表已经具有经度和纬度列，则可以使用类似以下内容的语句创建一个 ST\_Point 列，用于保存点的值：

```
ALTER TABLE my_table
  ADD point AS ST_Point(SRID=4326)
  COMPUTE( NEW ST_Point( longitude, latitude, 4326 ) );
```

- **使用构造程序和静态方法创建几何** - 可使用构造程序和静态方法来创建几何。

## 以图像形式查看空间数据 (Interactive SQL)

在 Interactive SQL 中，可以使用“空间预览”选项卡以图像形式查看几何，以便理解数据库中的数据所表示的内容。

### 前提条件

您必须对从中进行选择的表具有 SELECT 特权，或具有 SELECT ANY TABLE 系统特权。

### 过程

Interactive SQL 的各个实例与到数据库的不同连接相关联。从 Interactive SQL 打开“空间查看器”的某一实例时，“空间查看器”的该实例仍与 Interactive SQL 的该实例相关联并共享到数据库的连接。

当在“空间查看器”中执行查询时，如果尝试在与 Interactive SQL 相关联的实例中执行查询，可能会出现错误。同样，如果同时打开多个“空间查看器”实例，且这

些实例均由同一 **Interactive SQL** 实例创建，则一次只能有一个实例执行查询，其它实例必须待前一查询完成后才能执行查询。

---

**注意：**缺省情况下，**Interactive SQL** 将把“**结果**”窗口中的值截断为 256 个字符。如果 **Interactive SQL** 返回错误，则表示无法读取完整的列值，请增大截断值。要执行此操作，请单击“**工具**”»“**选项**”，然后在左侧窗格中单击“**SAP Sybase IQ**”。在“**结果**”选项卡中，将“**截断长度**”更改为一个更高的值，比如 5000。单击“**确定**”保存更改，再次执行查询，然后再次双击该行。

---

1. 使用 **Interactive SQL** 连接到数据库。
2. 执行一个查询，从表中选择空间数据。例如：

```
SELECT * FROM owner.spatial-table;
```
3. 在“**结果**”窗格中双击 **Shapes** 列中的任意一个值以在“**值**”窗口中打开该值。  
该值将以文本形式显示在“**值**”窗口的“**文本**”选项卡中。
4. 单击“**空间预览**”选项卡以可缩放矢量图形 (SVG) 的形式查看几何。

几何显示为可缩放矢量图形 (SVG)。

### 下一步

可以以几何形式查看空间数据，使用“**上一行**”和“**下一行**”按钮可查看结果集中其它值。

---

## 以图像形式查看空间数据（空间查看器）

---

可以使用 [空间查看器] 以图像形式查看多个几何，以便理解数据库中的数据所表示的内容。

### 前提条件

您必须对从中进行选择的表具有 **SELECT** 特权，或具有 **SELECT ANY TABLE** 系统特权。

### 过程

结果中行的顺序会影响图像在“**空间查看器**”中的显示方式，因为图像将按照行的处理顺序进行绘制，最后绘制的图像显示在最上方。结果集中后出现的形状较清晰，而先出现的形状会变得模糊不清。

1. 使用 **Interactive SQL** 连接到数据库，然后单击“**工具**”»“**空间查看器**”。
2. 在“**空间查看器**”的“**SQL**”窗格中执行一个如下所示的查询，然后单击“**执行**”：

```
SELECT * FROM GROUPO.SpatialShapes;
```

- 使用“**绘制轮廓多边形**”工具移除绘图中多边形的颜色，以显示所有形状的轮廓。此工具位于图像下方，在保存、缩放和平移控件的附近。

结果集中的所有几何以一个图像的形式显示在“**结果**”区中。

下一步

无。

## 从标准文本 (WKT) 文件装载空间数据

您可以通过使用标准文本文件 (WKT) 将空间数据添加到表中，此文件夹中包含可用于将空间数据装载到数据库中并能够以几何形式进行表示的文本。

### 前提条件

装载数据所需的特权取决于 `-gl` 服务器选项。如果 `-gl` 选项设置为 `ALL`，则必须满足以下条件之一：

- 您是表的所有者
- 您具有表的 `LOAD` 特权
- 您具有 `LOAD ANY TABLE` 系统特权
- 您具有 `ALTER ANY TABLE` 系统特权

如将 `-gl` 选项设置为 `DBA`，您必须具有 `LOAD ANY TABLE` 或 `ALTER ANY TABLE` 系统特权。

如果 `-gl` 选项设置为 `NONE`，则不允许执行 `LOAD TABLE`。

从客户端计算机上的文件进行装载时：

- 还需要 `READ CLIENT FILE` 特权。
- 需要具有对从中读取的目录的读取特权。
- 必须启用 `allow_read_client_file` 数据库选项。
- 必须启用 `read_client_file` 安全功能。

### 过程

- 创建一个包含空间数据的 `WKT` 文件，以便将其装载到数据库中。  
文件可以是 `LOAD TABLE` 语句支持的任何格式。
- 在 `Interactive SQL` 中连接到数据库。
- 使用与以下所示类似的语句创建一个表并将数据从文件装载到表中：

```
DROP TABLE IF EXISTS SA_WKT;
CREATE TABLE SA_WKT (
    description CHAR(24),
```

```
sample_geometry ST_Geometry(SRID=1000004326)
);

LOAD TABLE SA_WKT FROM 'C:\\Documents and Settings\\All Users\\
\\Documents\\SAP Sybase IQ 16\\Samples\\wktgeometries.csv'
DELIMITED BY ',';
```

这些数据即被装载到表中。

空间数据从 WKT 文件成功装载。

### 下一步

可以在 Interactive SQL 中使用 “空间查看器” 查看数据。

## 创建或管理空间参照系

---

使用 Interactive SQL 或 SAP Control Center 来创建和管理 SAP Sybase IQ 的空间参照测量单位。

要与 SRS 关联的测量单位必须已经存在。

借助 SAP Control Center，可以创建使用现有参照系作为模板的空间参照系 (SRS)，然后对设置进行编辑。因此，所选空间参照系必须与想要创建的空间参照系相似。

创建空间参照系需要以下特权之一：

- **MANAGE ANY SPATIAL OBJECT** 系统特权。
- **CREATE ANY OBJECT** 系统特权。

修改空间参照系需要以下特权之一：

- 您是空间参照系的所有者。
- 空间参照系的 **ALTER** 特权
- **MANAGE ANY SPATIAL OBJECT** 系统特权
- **ALTER ANY OBJECT** 系统特权。

删除空间参照系需要以下特权之一：

- **MANAGE ANY SPATIAL OBJECT** 系统特权。
- **DROP ANY OBJECT** 系统特权。
- 拥有空间参照系。

## 创建或管理空间测量单位

---

软件安装了多种测量单位。如果安装的测量单位不适合您的数据，则可以创建您自己的测量单位。

使用 **Interactive SQL** 或 **SAP Control Center** 来创建和管理 **SAP Sybase IQ** 的空间测量单位。

创建空间测量单位需要以下特权之一：

- **MANAGE ANY SPATIAL OBJECT** 系统特权。
- **CREATE ANY OBJECT** 系统特权。

删除空间测量单位需要以下特权之一：

- **MANAGE ANY SPATIAL OBJECT** 系统特权。
- **DROP ANY OBJECT** 系统特权。
- 您拥有空间测量单位。



# 高级空间主题

本节包含高级空间主题。

## 平面地球表示和球形地球表示的工作原理

---

SAP Sybase IQ 支持平面地球表示和球形地球表示。**平面地球**参照系将全部或部分地球表面投影到二维平面上，并使用简单的 2D 欧几里得几何。两点之间的线均为直线（除了圆弧线串），几何不能覆盖边缘（越过国际日期变更线）。

**球形地球**空间参照系使用椭圆柱体表示地球。根据计算将点映射到椭圆柱体，所有线均沿着最短路径和弧指向极点，几何可以越过国际日期变更线。

平面地球表示和球形地球表示都有自身的局限性。没有任何一个理想的地图投影可以最好地表示地球的所有要素，失真可能会影响物体的面积、形状、距离或方向，具体取决于该物体在地球上的位置。

### *球形地球空间参照系的局限性*

使用球形地球空间参照系（如 WGS 84）时，许多操作不可用。例如，计算距离限定为点或点集。

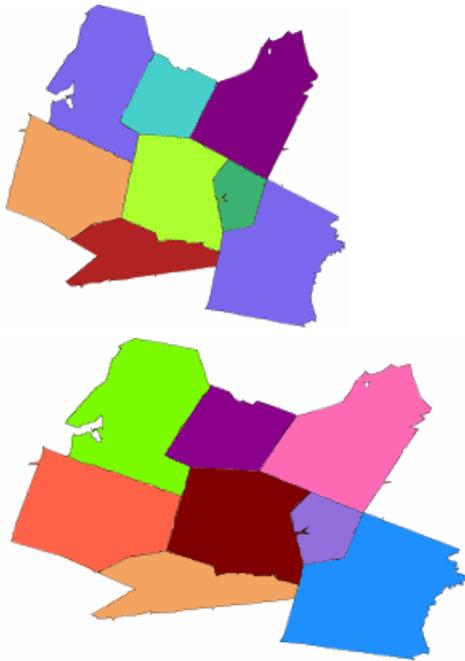
某些谓词和集合运算也不可用。

球形地球空间参照系中不允许使用圆弧线串。

球形地球空间参照系中的计算比平面地球空间参照系中的相应计算开销大。

### *平面地球空间参照系的局限性*

平面地球空间参照系是定义了投影的平面空间参照系。**投影**解决了平面地球空间参照系操作球形地球数据时出现的失真问题。有关如果不使用投影可能会出现失真的示例，以下两个图像显示马萨诸塞州的同一组邮政编码区域。第一个图像显示的是已专为马萨诸塞州数据进行投影的平面空间参照系 SRID 3586 中的数据。第二个图像显示的是尚未投影的平面空间参照系 (SRID 1000004326) 中的数据。在第二个图像中出现明显的失真，即距离、长度和面积均比实际值大，导致图像出现水平拉伸。



虽然在平面地球空间参照系中可以执行更多的计算，但由于投影的影响，只有有界区域的计算才准确。

处理几百千米的距离范围内的数据时，可以将球形地球数据投影到平面地球空间参照系后再执行距离计算，这样计算出来的结果精度会在合理范围内。要将数据投影到平面投影空间参照系，请使用 `ST_Transform` 方法。

## 对齐网格和公差如何影响空间计算

---

**对齐网格**是一种几何定位操作，用于使几何中的点与网格上的交叉点对齐。使点与**网格**对齐时，**X** 和 **Y** 值会有少量变化 - 类似于舍入。在空间数据的上下文中，网格是放置于空间参照系的二维表示上的线框。**SAP Sybase IQ** 使用方形网格。

作为一个简化的对齐网格示例，如果网格大小为 0.2，则从点 (14.2321, 28.3262) 到点 (15.3721, 27.1128) 的线将与从点 (14.2, 28.4) 到点 (15.4, 27.2) 的线相对齐。网格大小通常比此简化示例小得多，所以精度损失也会小得多。

缺省情况下，**SAP Sybase IQ** 自动设置网格大小，以便可以为空间参照系的 **X** 和 **Y** 界限内的每个点存储 12 位有效数字。例如，如果 **X** 的取值范围为 -180 到 180，**Y** 的取值范围为 -90 到 90，则数据库服务器会将网格大小设置为  $1e-9$  (0.000000001)。也就是说，水平和垂直网格线间的距离均为  $1e-9$ 。网格线的交叉点表示可以在空间参照系中表示的所有点。创建或装载几何时，各点的 **X** 坐标和 **Y** 坐标对齐到网格上最近的点。

**公差**定义一个距离，处于此距离之内的两点或几何部分被视为重合。这可以设想为：所有几何都表示为使用粗头划线工具绘制的点和线，其中线的粗度等于公差。使用粗头划线工具绘制时相接触的任何部分均视为在公差范围内相等。如果两点间的距离恰好等于公差，则认为在该公差范围二者不重合。

举个简单的公差示例，如果公差为 0.5，则点 (14.2, 28.4) 与点 (14.4, 28.2) 将被视为相等。因为这两点之间的距离（其单位与 X 和 Y 相同）约为 0.283，小于公差。但是，公差通常比此简化示例小得多。

公差可能会导致极小的几何变为无效。长度小于公差的线无效（因为其上的点均被视为相同）。同样，由所有在公差内相同的点构成的多边形也被视为无效。

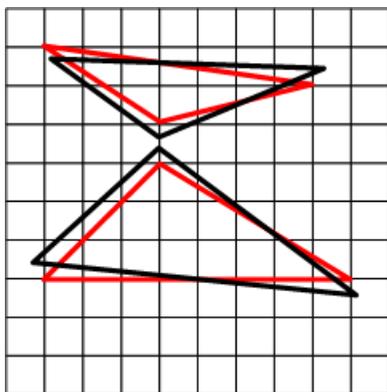
对齐网格和公差是在空间参照系上设置，并始终以与 X 和 Y（或经度和纬度）坐标相同的单位进行指定。对齐网格和公差共同作用，以克服不精确算术和不精确数据所导致的问题。但是应当注意其行为对空间操作结果的影响。

**注意：**对于平面空间参照系，建议不要将网格大小设置为 0，因为这会导致空间操作的结果不正确。对于球形地球空间参照系，网格大小和公差必须设置为 0。当执行球形地球操作时，SAP Sybase IQ 会对内部投影使用固定的网格大小和公差。

以下各示例说明了网格大小和公差设置对空间计算的影响。

### 示例 1：对齐网格影响相交结果

两个三角形（以黑色显示）装载到公差设置为网格大小的空间参照系，并且图中的网格基于该网格大小。红色三角形表示三角形顶端对齐到网格后的黑色三角形。注意，原始三角形（黑色）间的距离在公差范围内，但对齐后的红色三角形则不是。对于这两个几何，ST\_Intersects 返回 0。如果公差大于网格大小，那么对于这两个几何，ST\_Intersects 将返回 1。

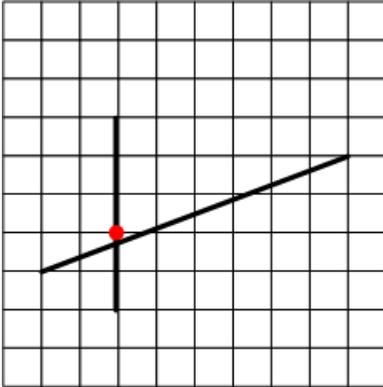


### 示例 2：公差影响相交结果

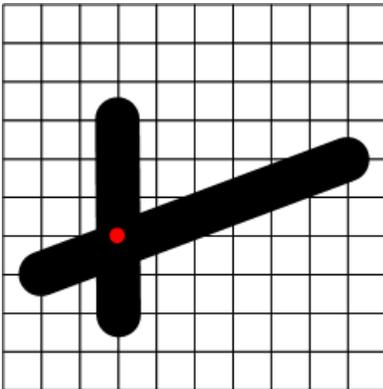
在下面的示例中，两条线位于公差设置为 0 的空间参照系中。这两条线的交点被对齐到网格中最近的顶点。由于公差设置为 0，因此确定两条线的交点是否与斜线相交的检测将返回假值。

也就是说，公差为 0 时，下面的表达式返回 0:

```
vertical_line.ST_Intersection( diagonal_line ).ST_Intersects( diagonal_line )
```

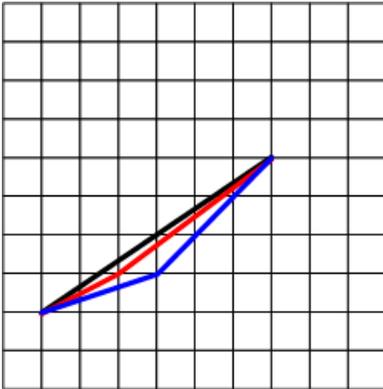


然而，将公差设置为网格大小（缺省设置）将使交点位于粗斜线内部。因此，对交点是否与斜线在公差范围内相交的测试将通过：



### 示例 3：公差和传递性

在使用了公差的空间计算中，传递性不一定会保持。例如，假设公差等于网格大小的空间参照系中有以下三条线：



`ST_Equals` 方法认为在该公差范围内黑色和红色线相同，红色和蓝色线也在公差范围内相同，但在该公差范围内黑色和蓝色线不相同。`ST_Equals` 没有传递性。

`ST_OrderingEquals` 认为这三条线各不相同，因此 `ST_OrderingEquals` 具有传递性。

#### 示例 4：网格和公差设置对不精确数据的影响

假设您的数据位于投影平面空间参照系中，此参照系 10 厘米范围内最为精确，并且在 10 米以内都是精确的。可有以下三个选择：

1. 使用 SAP Sybase IQ 所选择的缺省网格大小和公差，此值通常大于数据的精度。虽然这种方法可提供最大精度，但有些谓词（如 `ST_Intersects`、`ST_Touches` 和 `ST_Equals`）所给出的结果可能与有些几何的预期值不同，具体取决于几何值的精确度。例如，彼此共用一个边界的两个相邻多边形，如果左侧多边形的边界数据距右侧多边形的左侧几米远，则 `ST_Intersect` 可能不返回 `true`。
2. 将网格大小设置为小到足以表示最精确数据的值（在本例中为 10 厘米），且比公差至少小四倍；将公差设置为数据始终精确到的距离（在本例中为 10 米）。使用这种策略存储数据时不会失去精度，而且，即使数据仅精确到 10 米范围内，谓词也会给出预期结果。
3. 将网格大小和公差设置为数据的精度（在本例中为 10 米）。使用这种方法，数据将捕捉到数据精度范围以内，但对于精度高于 10 米的数据，则会失去该范围以外的精度。

多数情况下，谓词可给出预期结果；但在某些情况下不会给出预期结果。例如，如果两个点彼此之间的距离在 10 厘米以内，但位置接近网格交点的中点，则其中一个点会捕捉到一个方向，而另一个点则捕捉到另外一个方向，从而使两点间的距离接近 10 米。因此，在本例中不建议将网格大小和公差设置为与数据的精度匹配。

## 内插值如何影响空间计算

**内插值**是使用几何中的已知点来近似未知点的过程。当计算涉及圆弧时，有几个空间方法和谓词使用内插值。内插值将圆弧变成一连串直线。例如，表示四分之一弧的圆弧线串可被内插为具有 11 个控制点的线串。

### 内插值示例

1. 在 **Interactive SQL** 中，连接到示例数据库，执行以下语句，以便创建名为 `arc` 的用于存储圆弧线串的变量：

```
CREATE VARIABLE arc ST_CircularString;
```

2. 执行以下语句，以创建圆弧线串并将其存储在 `arc` 变量中：

```
SET arc = NEW ST_CircularString( 'CircularString( -1 0, -0.707107  
0.707107, 0 1 )' );
```

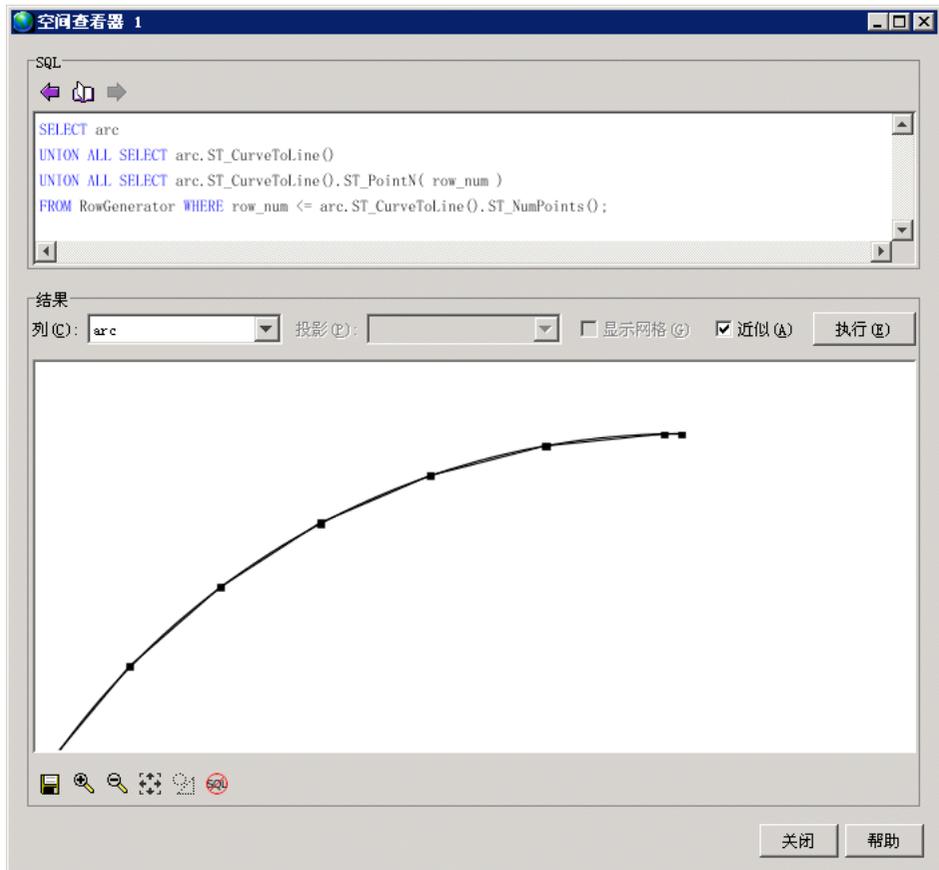
3. 执行以下语句，以使用 `st_geometry_interpolation` 选项将相对公差临时设置为 1%。

```
SET TEMPORARY OPTION st_geometry_interpolation = 'relative-  
tolerance-percent=1';
```

将相对公差设置为 1% 是可选操作，但可使内插值对本示例的目的的作用更加明显。

4. 打开 **“空间查看器”**（在 **Interactive SQL** 中，单击 **“工具”** » **“空间查看器”**），然后执行以下查询以查看圆弧线串：

```
SELECT arc  
UNION ALL SELECT arc.ST_CurveToLine()  
UNION ALL SELECT arc.ST_CurveToLine().ST_PointN( row_num )  
FROM RowGenerator WHERE row_num <=  
arc.ST_CurveToLine().ST_NumPoints();
```



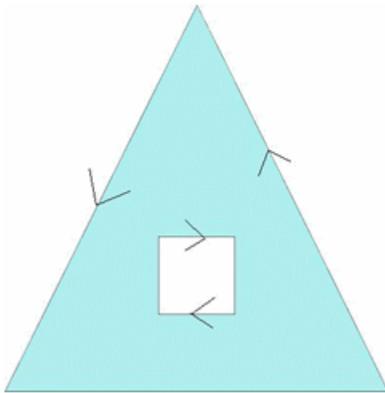
请注意如何将圆弧分成一连串线串。由于将相对公差设置为 1%，每条线段显示为相对于真弧线向内弯曲的直线。内插线串与真弧线之间的最大距离为圆弧半径的 1%。

## 多边形环方向的工作原理

在内部，SAP Sybase IQ 依据几何组分环的方向解释多边形。当按照定义的点顺序经过圆环时，多边形的内侧在环的左侧。PLANAR 和 ROUND EARTH 空间参照系也应用相同的规则。在大多数情况下，外环为逆时针方向，而内环为相反的（顺时针）方向。例外情况是 ROUND EARTH 中包含北极点或南极点的环。

缺省情况下，如果创建多边形时使用的环方向与 SAP Sybase IQ 内环方向不同，多边形将自动重新定向。使用 CREATE SPATIAL REFERENCE SYSTEM 语句的 POLYGON FORMAT 子句指定输入数据多边形环的方向。只有空间参照系的所有输入数据均使用相同的环方向时才能这样做。还可以在一些多边形和多表面构造函数上指定多边形格式。

例如，如果您创建多边形并按顺时针方向指定点 [Polygon((0 0, 5 10, 10 0, 0 0), (4 2, 4 4, 6 4, 6 2, 4 2))], 则数据库服务器会自动将点重新排列为逆时针旋转方向，如下所示: [Polygon((0 0, 10 0, 5 10, 0 0), (4 2, 4 4, 6 4, 6 2, 4 2))].



如果先指定内环后指定外环，则外环将显示为第一个环。

为了使多边形重定向在球形地球空间参照系中有效，多边形的直径被限定为 160°。

## 几何内部、几何外部和几何边界的工作原理

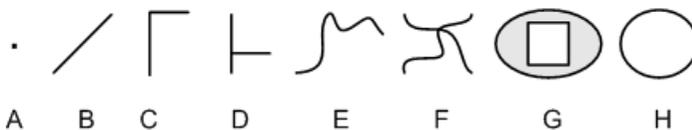
几何**内部**指作为几何组成部分的所有点，几何边界除外。

几何**外部**指不是几何组成部分的所有点。这可能包括内环内部的空间，例如：带孔的多边形。同样，线串环内部和外部的空间都被视为几何外部。

几何**边界**指通过 `ST_Boundary` 方法返回的部分。

了解几何边界有助于在对比两个几何时确定二者之间的关系。所有几何都具有内部和外部，但并非所有几何都具有边界，而且，并非所有几何的边界都始终是直观的。

以下是几何边界可能不直观的几种情况：



- **点** - 点（例如 A）没有边界。
- **线和曲线** - 线和曲线（B、C、D、E、F）的边界为它们的端点。几何 B、C 和 E 都有两个可以作为边界的端点。几何 D 有四个可以作为边界的端点，几何 F 也有四个。
- **多边形** - 多边形（例如 G）的边界为多边形的外环和所有内环。

- **环** - 环一起点和终点重合且不自相交的曲线（例如 **H**） - 没有边界。

## 空间比较的工作原理

您可通过两种方法来测试两个几何是否相等：**ST\_Equals** 和 **ST\_OrderingEquals**。这两种方法执行比较的方式不同，返回的结果也不同。

- **ST\_Equals** - 点的指定顺序并不重要，且进行点比较时会考虑公差。如果被比较的几何在公差范围内占用相同的空间，则将被视为相等几何。例如，如果两个线串占用相同的空间，即使其中一个使用较多的点进行定义，则它们仍被视为相等。
- **ST\_OrderingEquals** - 使用 **ST\_OrderingEquals** 时，**ST\_OrderingEquals** 下被视为相等的两个几何必须包含相同的对象层次，其中的点必须完全相同且按同一顺序排列。也就是说，这两个几何必须完全相同。

为了说明 **ST\_Equals** 和 **ST\_OrderingEquals** 在比较结果上的差异，我们将以下面的线为例。**ST\_Equals** 会将这些线全部视为相等（假定线 C 在公差范围内）。而 **ST\_OrderingEquals** 则会将这些线视为彼此不相等。

1 ● ——— A ——— ● 2	LineString( 0.0 0.0, 4.0 0.0 )
2 ● ——— B ——— ● 1	LineString( 4.0 0.0, 0.0 0.0 )
1 ● ——— C ——— ● 2	LineString( 0.0 0.0, 4.0 0.000001 )
1 ● ——— D ——— ● 3 2	LineString( 0.0 0.0, 1.0 0.0, 4.0 0.0 )
1 ● ——— E ——— ● 2	MultiLineString(( 0.0 0.0, 4.0 0.0 ))

### SAP Sybase IQ 执行几何比较的方式

数据库服务器使用 **ST\_OrderingEquals** 执行诸如 **GROUP BY** 和 **DISTINCT** 之类的比较。

例如，处理以下查询时，如果两个形状表达式使 **ST\_OrderingEquals() = 1**，则服务器认为两行相等：

```
SELECT DISTINCT Shape FROM GROUPO.SpatialShapes;
```

SQL 语句比较两个几何时可以使用 "等于" 运算符 (=) 或 "不等于" 运算符 (<> 或 !=)，包括使用带有子查询和关键字 **ANY** 或 **ALL** 的搜索条件。也可以在 **IN** 搜索条件中使用几何。例如，**geom1 IN (geom-expr1, geom-expr2, geom-expr3)**。对于所有这些搜索条件，都将使用 **ST\_OrderingEquals** 语义判断是否相等。

不能使用其它比较运算符来确定一个几何是否小于或大于另一个几何（例如，不可以使用 **geom1 < geom2**）。也就是说，**ORDER BY** 子句中不能包含几何表达式。但是，可以测试成员是否属于某个集合。

## 空间关系的工作原理

为获得最优性能，应使用 `ST_Within` 或 `ST_Touches` 等方法测试几何间的单个特定关系。但如果要测试的关系不止一个，则最好选用 `ST_Relate` 方法，因为该方法可同时测试多个关系。此外，希望针对谓语的不同解释进行测试时，`ST_Relate` 同样也是一个不错的方法。

通常情况下，`ST_Relate` 被用作谓语，由用户指定要测试的具体关系。但 `ST_Relate` 也可用于确定两个几何间的所有可能关系。

### 将 `ST_Relate` 用作谓语

`ST_Relate` 通过对几何内部、边界和外部执行**相交测试**来评估几何间的关系。然后用包括 9 个字符的 **DE-9IM**（维度扩展 9 交集模型）格式的字符串来描述几何间的关系，其中字符串的每个字符都表示相交测试的结果的维度。

将 `ST_Relate` 用作谓语时，需要传递 **DE-9IM** 字符串，以反映要测试的相交结果。如果几何满足 **DE-9IM** 字符中所指定的条件，则 `ST_Relate` 将返回 **1**。如果不满足，则返回 **0**。如果任一几何或两个几何都为 **NULL**，则返回 **NULL**。

由 9 个字符组成的 **DE-9IM** 字符串是对成对矩阵的展平表示，该成对矩阵由内部、边界和外部间的相交测试组成。下表按测试的执行顺序（从左到右、从上到下）显示了 9 个相交测试：

	<b>g2 内部</b>	<b>g2 边界</b>	<b>g2 外部</b>
<b>g1 内部</b>	Interior(g1) ∩ Interior(g2)	Interior(g1) ∩ Boundary(g2)	Interior(g1) ∩ Exterior(g2)
<b>g1 边界</b>	Boundary(g1) ∩ Interior(g2)	Boundary(g1) ∩ Boundary(g2)	Boundary(g1) ∩ Exterior(g2)
<b>g1 外部</b>	Exterior(g1) ∩ Interior(g2)	Exterior(g1) ∩ Boundary(g2)	Exterior(g1) ∩ Exterior(g2)

指定 **DE-9IM** 字符串时，9 个字符中的任何一个字符都可被指定为 \*、0、1、2、T、或 F。这些值表示通过相交创建的几何维数。

指定的值：	相交测试结果应返回的值：
T	以下值之一：0、1、2（与任何维度相交）
F	-1
*	-1、0、1、2（任何值）
0	0

指定的值:	相交测试结果应返回的值:
1	1
2	2

假设您想用 `ST_Relate` 方法和自定义 DE-9IM 字符串作为 "within" 谓语句来测试某一几何是否位于另一几何中:

```
SELECT new ST_Polygon('Polygon(( 2 3, 8 3, 4 8, 2
3 ))').ST_Relate( new ST_Polygon('Polygon((-3 3, 3 3, 3 6, -3 6, -3
3))'), 'T*F**F***' );
```

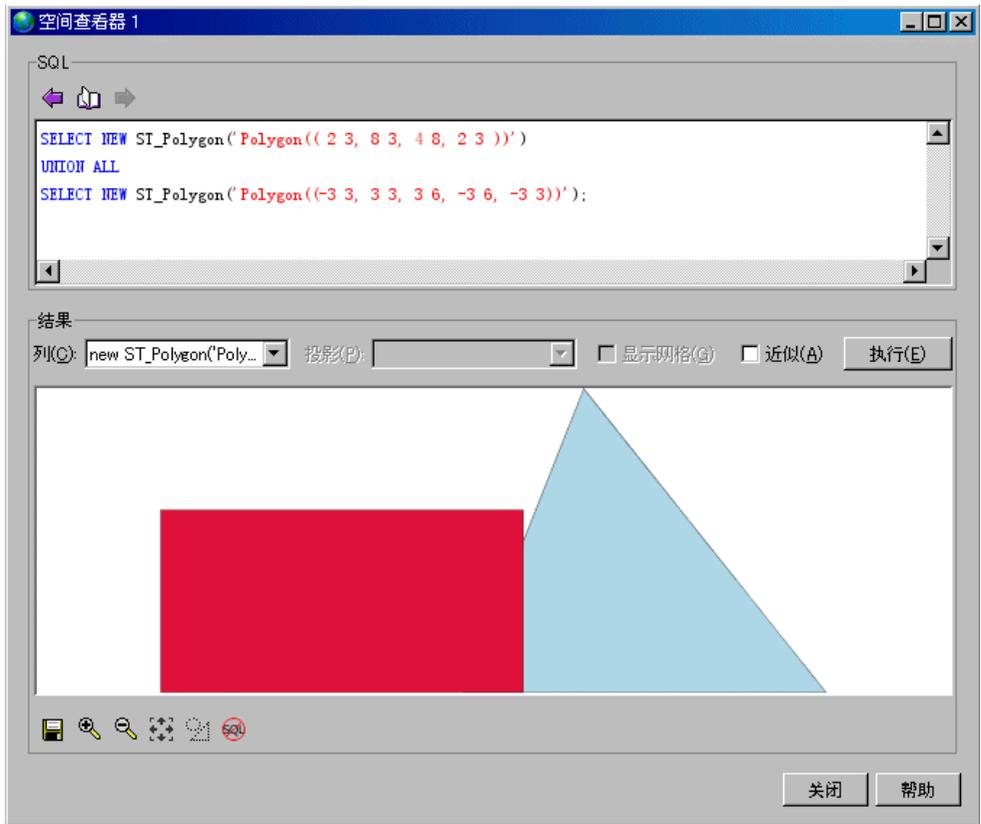
这等同于让 `ST_Relate` 在执行相交测试时寻找以下条件:

	g2 内部	g2 边界	g2 外部
g1 内部	以下值之一: <b>0, 1, 2</b>	以下值之一: 0, 1, 2, -1	<b>-1</b>
g1 边界	以下值之一: 0, 1, 2, -1	以下值之一: 0, 1, 2, -1	<b>-1</b>
g1 外部	以下值之一: 0, 1, 2, -1	以下值之一: 0, 1, 2, -1	以下值之一: 0, 1, 2, -1

但执行查询时, `ST_Relate` 将返回 **0**, 表示第一个几何不在第二个几何中。

要查看两个几何并将其外观与所测试的关系进行比较, 请在 **Interactive SQL 空间查看器** (“工具” » “空间查看器”) 中执行以下语句:

```
SELECT NEW ST_Polygon('Polygon(( 2 3, 8 3, 4 8, 2 3 ))')
UNION ALL
SELECT NEW ST_Polygon('Polygon((-3 3, 3 3, 3 6, -3 6, -3 3))');
```



*将 ST\_Relate 用作非谓语句*

将 ST\_Relate 用作非谓语句将返回两个几何间的完整关系。

例如，假设您有两个与前一示例中所使用的几何完全相同的几何，而且您想知道这两个几何之间的关系。那么，您应该在 Interactive SQL 中执行以下语句以返回定义这两个几何间关系的 DE-9IM 字符串。

```
SELECT new ST_Polygon('Polygon(( 2 3, 8 3, 4 8, 2 3 )))'.ST_Relate(new ST_Polygon('Polygon((-3 3, 3 3, 3 6, -3 6, -3 3 )))');
```

ST\_Relate 返回的 DE-9IM 字符串为 [212111212]。

该值的矩阵视图表明两个几何间有多个相交点：

	g2 内部	g2 边界	g2 外部
g1 内部	2	1	2
g1 边界	1	1	1

g1 外部	2	1	2
-------	---	---	---

## 空间维度的工作原理

---

几何子类型在各自具有独特属性的同时还从 `ST_Geometry` 超类型继承属性。几何子类型具有以下维数值之一：

- **-1** - 值为 -1 表示为空几何（不包含任何点）。
- **0** - 值为 0 表示几何没有长度和面积。子类型 `ST_Point` 和 `ST_MultiPoint` 的维数值为 0。点表示可以用单一坐标对表示的几何要素，一组不相连的点表示多点要素。
- **1** - 值为 1 表示几何有长度但没有面积。维数值为 1 的子类型集是 `ST_Curve` (`ST_LineString`、`ST_CircularString` 和 `ST_CompoundCurve`) 的子类型或包含这些类型的集合类型，但没有表面。在 GIS 数据中，维数值为 1 的几何用于定义线性要素，如溪流、河流分支系统和路段。
- **2** - 值为 2 表示几何具有面积。维数值为 2 的子类型集是 `ST_Surface` (`ST_Polygon` 和 `ST_CurvePolygon`) 的子类型或包含这些类型的集合类型。多边形和多面表示具有周长（用于将所定义的区域围起来）的几何要素，如湖泊或公园。

几何的维度与几何中每个点的坐标维度数无关。

单个 `ST_GeomCollection` 可包含维数不同的几何，且将返回维数最高的几何



# 教程：试用空间要素

本教程将指导您在 SAP Sybase IQ 中试用某些空间要素。要试用空间要素，首先请将 ESRI shapefile 装载到示例数据库 (demo.db) 中以获得可供试用的有效空间数据。

教程分为以下几部分：

## 特权

要执行此教程，必须具有以下特权：

MANAGE ANY SPATIAL OBJECT 系统特权

CREATE TABLE 系统特权

WRITE FILE 系统特权

对 GROUPO.SpatialContacts 表的 SELECT 特权

## 第 1 课：安装附加测量单位和空间参照系

本课将指导您使用 sa\_install\_feature 系统过程安装大量预定义的测量单位和空间参照系，在本教程后面的部分中将会用到它们。

### 前提条件

本课假定您拥有在本教程（Tutorial: Experimenting with the spatial features）开头的“特权”部分中列出的角色和特权。

### 过程

1. 使用 Interactive SQL，启动并连接到示例数据库 (demo.db)。

示例数据库位于 %IQDIRSAMP16% 中。

2. 在 Interactive SQL 中执行以下语句：

```
CALL sa_install_feature( 'st_geometry_predefined_srs' );
```

语句执行完成后，即成功安装了附加的测量单位和空间参照系。

3. 要确定数据库中所安装的测量单位，请在 Interactive SQL 中执行以下查询：

```
SELECT * FROM ST_UNITS_OF_MEASURE;
```

4. 要确定数据库中所安装的空间参照系，在 Interactive SQL 中执行以下查询：

```
SELECT * FROM ST_SPATIAL_REFERENCE_SYSTEMS;
```

将返回已安装空间参照系的列表。

## 第 2 课：下载 ESRI shapefile 数据

---

### 前提条件

本课假定您已完成前面的所有课程。请参见 Lesson 1: Install additional units of measure and spatial reference systems。

本课假定您拥有在本教程（Tutorial: Experimenting with the spatial features）开头的“特权”部分中列出的角色和特权。

### 过程

1. 创建名为 `c:\temp\massdata` 的本地目录。
2. 转至以下 URL：<http://www2.census.gov/cgi-bin/shapefiles2009/national-files>
3. 在页面右侧的“**State- and County-based Shapefiles**”（基于州和县的 Shapefile）下拉列表中，选择“**Massachusetts**”（马萨诸塞州），然后单击“**Submit**”（提交）。
4. 在页面左侧单击“**5-Digit ZIP Code Tabulation Area (2002)**”（5 位邮政编码汇总地区 (2002)），然后单击“**Download Selected Files**”（下载所选文件）。
5. 当出现提示时，将 zip 文件 `multiple_tiger_files.zip` 保存到 `c:\temp\massdata` 中，并解压缩其内容。这将创建一个名为 `25_MASSACHUSETTS` 的子目录，其中包含另一个名为 `t1_2009_25_zcta5.zip` 的 zip 文件。
6. 将 `t1_2009_25_zcta5.zip` 中的内容解压到 `C:\temp\massdata`。

本课将解压缩出五个文件，包括可用于向数据库装载空间数据的 ESRI shapefile (.shp)。

## 第 3 课：装载 ESRI shapefile 数据

---

本课将指导您如何确定 ESRI shapefile 中的列，然后使用该信息创建将装载数据的表。

### 前提条件

本课假定您已完成前面的所有课程。请参见 Lesson 1: Install additional units of measure and spatial reference systems。

本课假定您拥有在本教程（Tutorial: Experimenting with the spatial features）开头的“特权”部分中列出的角色和特权。

## 过程

如果因特权问题而导致在运行任一步骤时遇到困难，请咨询管理员 `-gl` 数据库选项所设置的值，然后阅读 `st_geometry_load_shapefile` 系统过程的特权部分，以确定所需的相应特权。

1. 由于空间数据与特定空间参照系相关联，因此向数据库中装载数据时，必须将其装载到同一空间参照系或者至少具有等效定义的空间参照系中。要查找 **ESRI shapefile** 的空间参照系信息，请在文本编辑器中打开项目文件 `c:\temp\massdata\tl_2009_25_zcta5.prj`。此文件中包含所需的空间参照系信息。

```
GEOGCS["GCS_North_American_1983", DATUM["D_North_American_1983",
SPHEROID["GRS_1980",6378137,298.257222101]],
PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]]
```

字符串 **GCS\_North\_American\_1983** 即为与数据相关联的空间参照系的名称。

2. 对 **ST\_SPATIAL\_REFERENCE\_SYSTEMS** 视图执行快速查询后 (`SELECT * FROM ST_SPATIAL_REFERENCE_SYSTEMS WHERE srs_name='GCS_North_American_1983';`)，查询结果显示此名称并未出现在预定义 **SRS** 列表中。但您可以查询具有相同定义的空间参照系并改为使用该空间参照系：

```
SELECT *
FROM ST_SPATIAL_REFERENCE_SYSTEMS
WHERE definition LIKE '%1983%'
AND definition LIKE 'GEOGCS%';
```

该查询返回单个空间参照系，即 **SRID** 为 **4269** 的 **NAD83**，该空间参照系具有相同的定义。这是将分配给从 **shapefile** 装载的数据的 **SRID**。

3. 在 **Interactive SQL** 中，执行以下语句，以创建名为 **Massdata** 的表，将 **shapefile** 装载到表中，将 **SRID 4269** 指派给数据。装载约需一分钟。

```
CALL st_geometry_load_shapefile ( 'c:\\temp\\massdata\\
\tl_2009_25_zcta5.shp',
4269,
'Massdata' );
```

---

**注意：**“导入向导”也支持从 **shapefile** 装载数据。

---

4. 在 **Interactive SQL** 中，查询该表以查看过去位于 **shapefile** 中的数据：

```
SELECT * FROM Massdata;
```

结果中的每一行代表一个邮政编码地区的数据。

**geometry** 列以多边形（一个地区）或多面（两个或多个不相邻的地区）的形式保存邮政编码地区的形状信息。

5. **ZCTA5CE** 列保存邮政编码。为了在后面的教程中更容易查阅此列，请在 **Interactive SQL** 中执行以下 **ALTER TABLE** 语句，以将列名更改为 **[ZIP]**：

```
ALTER TABLE Massdata
RENAME ZCTA5CE TO ZIP;
```

6. INTPTLON 和 INTPTLAT 两列分别表示邮政编码地区的中心点的 X 和 Y 坐标。在 Interactive SQL 中执行以下 ALTER TABLE 语句，以创建名为 CenterPoint 类型的 ST\_Point 列，然后将每组 X 和 Y 变成 CenterPoint 中的值。

```
ALTER TABLE Massdata  
ADD CenterPoint AS ST_Point(SRID=4269)  
COMPUTE( new ST_Point( CAST( INTPTLON AS DOUBLE ), CAST( INTPTLAT  
AS DOUBLE ), 4269 ) );
```

现在，Massdata.CenterPoint 中的每个 ST\_Point 值表示存储在 Massdata.geometry 中的邮政编码地区的中心点。

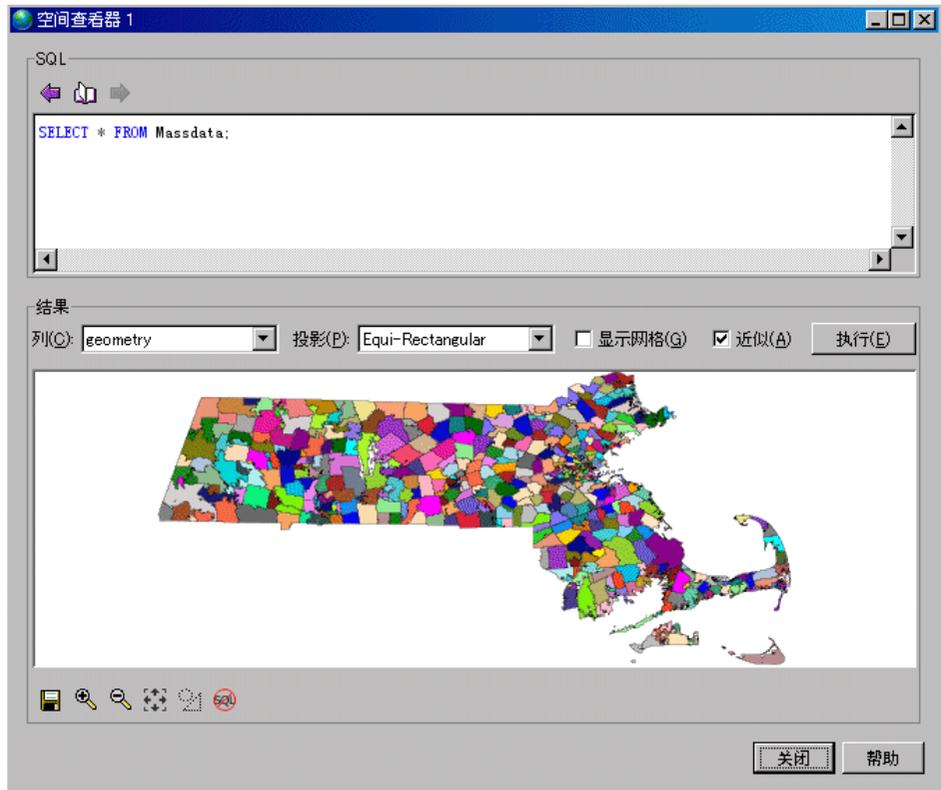
7. 要将单个几何（一个邮政编码地区）作为形状进行查看，请双击 Massdata.geometry 中除了第一个值以外的任何一个值，然后单击“列值”窗口中的“空间预览”选项卡。

如果收到的错误消息指出值过大或建议您在结果中包括主键，那是因为为了在 Interactive SQL 中显示这些值，已将它们截断。要修复此错误，您可以修改查询以在结果中包括主键列，也可以调整 Interactive SQL 的“截断长度”设置。如果您不想每次查询几何时都必须在结果中包括主键以在 Interactive SQL 中查看几何，那么建议您更改 [截断长度] 设置。

要更改 Interactive SQL 的“截断长度”设置，请单击“工具”»“选项”»“SAP Sybase IQ”，然后将“截断长度”设置成较大的数字，如 100000。

8. 要将整个数据集作为一个形状进行查看，请单击“工具”»“空间查看器”打开 SAP Sybase IQ “空间查看器”，然后在 Interactive SQL 中执行以下查询：

```
SELECT geometry FROM Massdata  
UNION ALL SELECT CenterPoint FROM Massdata;
```



ESRI shapefile 数据已装载。

## 第 4 课：查询空间数据

本课将以切合实际的示例来指导您使用某些空间方法查询数据。还将学习如何计算距离，其中将涉及向数据库中添加测量单位。

### 前提条件

本课假定您已完成前面的所有课程。请参见 Lesson 1: Install additional units of measure and spatial reference systems。

本课假定您拥有在本教程（Tutorial: Experimenting with the spatial features）开头的“特权”部分中列出的角色和特权。

### 过程

对 SpatialContacts 和 Massdata 表中的一个或两个执行查询。SpatialContacts 已存在于数据库中，用于保存人们的姓名和联系信息—其中许多人住在马萨诸塞州。

1. 在 **Interactive SQL** 中，创建名为 [`@Mass_01775`] 的变量，用于保存邮政编码地区 `01775` 的关联几何。

```
CREATE VARIABLE @Mass_01775 ST_Geometry;  
SELECT geometry INTO @Mass_01775  
FROM Massdata  
WHERE ZIP = '01775';
```

2. 假设您想要在 **SpatialContacts** 表中查找邮政编码地区 `01775` 及其周边邮政编码地区中的所有联系人信息。要达到这个目的，可以使用 **ST\_Intersects** 方法，该方法可返回与指定的几何相交或与其完全相同的几何。您将在 **Interactive SQL** 中执行以下语句：

```
SELECT c.Surname, c.GivenName, c.Street, c.City, c.PostalCode,  
z.geometry  
FROM Massdata z, GROUPO.SpatialContacts c  
WHERE  
c.PostalCode = z.ZIP  
AND z.geometry.ST_Intersects( @Mass_01775 ) = 1;
```

3. **Massdata.geometry** 中的所有行都与同一空间参照系 (**SRID 4269**) 相关联，这是因为创建几何列并将数据装载到其中时为列分配了 **SRID 4269**。

但您仍可以创建**未声明的 ST\_Geometry** 列（即，未指派 **SRID** 的列）。有时，您可能需要在单个列中存储与不同 **SRS** 相关联的空间值。此时，在对这些值进行运算时，将分别使用与各个值相关联的空间参照系。

不过，使用未声明的列存在着一个风险，即：数据库服务器不会阻止您更改与未声明列中的数据相关联的空间参照系。

但如果列已具有声明的 **SRID**，则数据库服务器将不允许您修改与其中的数据相关联的空间参照系。必须先卸载数据，再截断声明列中的数据，然后再更改空间参照系，最后再重新装载数据。

无论列声明与否，都可通过 **ST\_SRID** 方法来确定与列中的值相关联的 **SRID**。例如，以下语句可显示指派给 **Massdata.geometry** 列中各行的 **SRID**：

```
SELECT geometry.ST_SRID()  
FROM Massdata;
```

4. 可以使用 **ST\_CoveredBy** 方法来检查某一几何是否完全位于另一几何中。例如，**Massdata.CenterPoint** (**T\_Point** 类型) 包含邮政编码地区中心的经度/纬度坐标，而 **Massdata.geometry** 则包含反映邮政编码地区的多边形。可通过在 **Interactive SQL** 中执行以下查询进行快速检查，来确保所有 **CenterPoint** 值均已设置在其邮政编码地区之内：

```
SELECT * FROM Massdata  
WHERE NOT (CenterPoint.ST_CoveredBy(geometry) = 1);
```

未返回任何行，说明所有 **CenterPoint** 值都包含在 **Massdata.geometry** 中的对应相关几何中，当然，此项检查并不会验证这些值是否为真正的中心。如果想执行此验证，需要将数据投影到平面地球空间参照系，然后使用 **ST\_Centroid** 方法检查 **CenterPoint** 值。稍后将在本教程中介绍投影内容。

5. 可以使用 `ST_Distance` 方法测量两个邮政编码地区中心点间的距离。例如，假设您希望得到邮政编码地区 01775 方圆 100 英里之内的邮政编码列表，则可在 **Interactive SQL** 中执行以下查询：

```
SELECT c.PostalCode, c.City,
       z.CenterPoint.ST_Distance( ( SELECT CenterPoint
                                   FROM Massdata WHERE ZIP = '01775' ),
                                   'Statute mile' ) dist,
       z.CenterPoint
FROM Massdata z, GROUPO.SpatialContacts c
WHERE c.PostalCode = z.ZIP
      AND dist <= 100
ORDER BY dist;
```

6. 如果是否知道确切的距离并不重要，则可改用 `ST_WithinDistance` 方法来构造查询，此方法对于某些数据集（尤其是对于大型几何）的性能更佳：

```
SELECT c.PostalCode, c.City, z.CenterPoint
FROM Massdata z, GROUPO.SpatialContacts c
WHERE c.PostalCode = z.ZIP
      AND z.CenterPoint.ST_WithinDistance( ( SELECT CenterPoint
                                             FROM Massdata WHERE ZIP = '01775' ),
                                             100, 'Statute mile' ) = 1
ORDER BY c.PostalCode;
```

已对空间数据执行查询。

## 第 5 课：将空间数据输出为 SVG

在本课中，将创建一个 SVG 文档以查看以 WKT 表示的多面。可以导出 SVG 格式的几何，以在 **Interactive SQL** 或与 SVG 兼容的应用程序中查看该几何。

### 前提条件

本课假定您已完成前面的所有课程。请参见 [Lesson 1: Install additional units of measure and spatial reference systems](#)。

本课假定您拥有在本教程（[Tutorial: Experimenting with the spatial features](#)）开头的“特权”部分中列出的角色和特权。

### 过程

1. 在 **Interactive SQL** 中，执行以下语句，为示例几何创建一个变量：

```
CREATE OR REPLACE VARIABLE @svg_geom
ST_Polygon = (NEW ST_Polygon('Polygon ((1 1, 5 1, 5 5, 1 5, 1 1),
(2 2, 2 3, 3 3, 3 2, 2 2))'));;
```

2. 在 **Interactive SQL** 中，执行以下 `SELECT` 语句以调用 `ST_AsSVG` 方法：

```
SELECT @svg_geom.ST_AsSVG() AS svg;
```

结果集中只有一行，该行即为 SVG 图像。可以在 Interactive SQL 中使用“SVG 预览”功能查看该图像。要执行此操作，请双击结果行，然后选择“SVG 预览”选项卡。您应当会看到一个方形几何位于另一个方形几何内。

**注意：** 缺省情况下，Interactive SQL 将把“结果”窗口中的值截断为 256 个字符。如果 Interactive SQL 返回错误，则表示无法读取完整的列值，请增大截断值。要执行此操作，请单击“工具”»“选项”，然后在左侧窗格中单击“SAP Sybase IQ”。在“结果”选项卡中，将“截断长度”更改为一个更高的值，比如 5000。单击“确定”保存更改，再次执行查询，然后再次双击该行。

- 上一步骤介绍了如何在 Interactive SQL 中预览 SVG 图像。但是，将得到的 SVG 写入到一个文件中可能更实用，因为这样外部应用程序便可读取该文件。可以使用 `xp_write_file` 系统过程或 `WRITE_CLIENT_FILE` 函数 [String] 将 SVG 写入到相对于数据库服务器或客户端计算机的文件中。在下一示例中，将使用 `OUTPUT` 语句 [Interactive SQL]。

在 Interactive SQL 中，执行以下 `SELECT` 语句以调用 `ST_AsSVG` 方法并将几何输出到名为 `myPolygon.svg` 的文件中：

```
SELECT @svg_geom.ST_AsSVG();
OUTPUT TO 'c:\\temp\\massdata\\myPolygon.svg'
QUOTE ''
ESCAPES OFF
FORMAT TEXT
```

必须包括 `QUOTE ''` 和 `ESCAPES OFF` 子句，否则 XML 中将被插入回车符和单引号以保留空格，而这会导致输出的 SVG 文件无效。

- 在 Web 浏览器或支持查看 SVG 图像的应用程序中打开该 SVG 文件。也可以在文本编辑器中打开该 SVG，查看几何的 XML。
- `ST_AsSVG` 方法可从单个几何生成 SVG 图像。某些情况下，您希望生成包括组中所有 `shape` 的 SVG 图像。`ST_AsSVGAggr` 方法正是这样一个可以将多个几何合并到单个 SVG 图像中的集合函数。首先使用 Interactive SQL 创建一个变量用以保存 SVG 图像，然后使用 `ST_AsSVGAggr` 方法生成 SVG 图像。

```
CREATE OR REPLACE VARIABLE @svg XML;
SELECT ST_Geometry::ST_AsSVGAggr( geometry,
'attribute=fill="black"' )
INTO @svg
FROM Massdata;
```

此时，`@svg` 变量即保存了表示 `Massdata` 表中所有邮政编码地区的 SVG 图像。`'attribute=fill="black"'` 用于指定所生成图像的填充颜色。如果不指定，则数据库服务器将随机选择一种填充颜色。现在，您已经具有包含所需 SVG 图像的变量，接下来您便可将其写入到文件中，以便用其它应用程序进行查看。在 Interactive SQL 中执行以下语句，将 SVG 图像写入到相对于数据库服务器的文件中。

```
CALL xp_write_file( 'c:\\temp\\Massdata.svg', @svg );
```

也可以用 `WRITE_CLIENT_FILE` 函数写入相对于客户端应用程序的文件，但可能需要执行附加步骤才能确保启用了相应特权。如果在支持 SVG 数据的应用程序中打开该 SVG 图像，则会显示如下所示的图像：



该图像并未均匀地呈现黑色；相邻邮政编码地区的边界间存在细小的间隙。这些间隙实际上是各几何间的白线，这也是渲染 SVG 的特点。而数据间其实并不存在任何间隙。较大的白线是河流和湖泊。

该几何已被视为 SVG。

## 第 6 课：投影空间数据

---

本课指导您如何将数据投影到使用平面地球模型的空间参照系中，以便计算面积和距离测量值。

### 前提条件

本课假定您已完成前面的所有课程。请参见 [Lesson 1: Install additional units of measure and spatial reference systems](#)。

本课假定您拥有在本教程（[Tutorial: Experimenting with the spatial features](#)）开头的“特权”部分中列出的角色和特权。

### 过程

从 ESRI shapefile 向数据库中装载数据时，已为 Massdata 中的空间值指派了 SRID 4269 (NAD83 空间参照系)。SRID 4269 是球形地球空间参照系。然而，某些计算操作（例如，计算几何面积）和某些空间谓词在球形地球模型中不被支持。如果数据当前与球形地球空间参照系相关联，则可创建一个用于将数值投影到平面地球空间参照系的新空间列，然后对该列执行计算操作。

1. 要测量表示邮政编码地区的多边形的面积，必须先将 `Massdata.geometry` 中的数据投影到平面地球空间参照系。

要选择合适的 `SRID` 以将 `Massdata.geometry` 中的数据投影到其中，请使用 `Interactive SQL` 按如下方式查询 `ST_SPATIAL_REFERENCE_SYSTEMS` 统一视图，以获得包含 "马萨诸塞州" 字样的 `SRID`：

```
SELECT * FROM ST_SPATIAL_REFERENCE_SYSTEMS WHERE srs_name LIKE '%massachusetts%';
```

此查询可返回几个适用于马萨诸塞州数据的 `SRID`。本教程中将使用 `3586`。

2. 现在，必须创建一个名为 `Massdata.proj_geometry` 的列，然后用 `ST_Transform` 方法将几何投影到该列，进而投影到 `3586` 中。为此，请在 `Interactive SQL` 中执行以下语句：

```
ALTER TABLE Massdata
ADD proj_geometry
AS ST_Geometry(SRID=3586)
COMPUTE( geometry.ST_Transform( 3586 ) );
```

3. 可以使用 `Massdata.proj_geometry` 计算面积。例如，在 `Interactive SQL` 中执行以下语句：

```
SELECT zip, proj_geometry.ST_ToMultiPolygon().ST_Area('Statute
Mile') AS area
FROM Massdata
ORDER BY area DESC;
```

---

**注意：**球形地球空间参照系不支持 `ST_Area`，但支持 `ST_Distance`，不过，仅限于计算点几何之间的距离。

---

4. 要了解投影至另一空间参照系对距离计算的影响，可使用以下查询在球形地球模型（更精确）或投影平面地球模型上计算邮政编码地区中心点间的距离。两种模型所得出的计算结果毫无差异，因为所选投影适合该数据集。

```
SELECT M1.zip, M2.zip,
       M1.CenterPoint.ST_Distance( M2.CenterPoint, 'Statute
Mile' ) dist_round_earth,

M1.CenterPoint.ST_Transform( 3586 ).ST_Distance( M2.CenterPoint.S
T_Transform( 3586 ),
       'Statute Mile' ) dist_flat_earth
FROM Massdata M1, Massdata M2
WHERE M1.ZIP = '01775'
ORDER BY dist_round_earth DESC;
```

5. 假设您想查找与邮政编码地区 `01775` 相毗邻的邮政编码地区。要达到此目的，需要使用 `ST_Touches` 方法。`ST_Touches` 方法用于对几何进行比较，以了解某一几何是否与另一几何相接触但不与之相重叠。`ST_Touches` 返回的结果不包括与邮政编码 `01775` 相对应的行（这不同于 `ST_Intersects` 方法）。

```
CREATE OR REPLACE VARIABLE @Mass_01775 ST_Geometry;
SELECT geometry INTO @Mass_01775
FROM Massdata
WHERE ZIP = '01775';
```

```
SELECT record_number, proj_geometry
FROM Massdata
WHERE
proj_geometry.ST_Touches( @Mass_01775.ST_Transform( 3586 ) ) = 1;
```

6. 可以使用 **ST\_UnionAggr** 方法返回表示一组邮政编码地区联合的几何。例如，假设您想获得一个反映 01775 所毗邻的邮政编码区域（但不包括 01775 区域）联合的几何。

在 **Interactive SQL** 中，单击 **“工具”** » **“空间查看器”**，然后执行以下查询：

```
SELECT ST_Geometry::ST_UnionAggr(proj_geometry)
FROM Massdata
WHERE
proj_geometry.ST_Touches( @Mass_01775.ST_Transform( 3586 ) ) = 1;
```

双击结果即可对其进行查看。

如果收到错误消息，指出无法从数据库读取整列内容，则需要增大 **Interactive SQL** 的 **“截断长度”** 设定值。要执行此操作，请在 **Interactive SQL** 中单击 **“工具”** » **“选项”** » **“SAP Sybase IQ”**，然后将 **“截断长度”** 设置成较大的数字。然后再次执行查询并查看几何。

已完成本教程。



# 访问和操作空间数据

本节介绍访问、操作和分析空间数据时可以使用的类型、方法和构造函数。可将空间数据类型视为数据类型或类。每个空间数据类型都有关联的方法和构造函数，您可以使用它们访问数据。



## 附录 - SQL 语句

本文中提及的 SQL 语句的参考资料。

### CREATE SPATIAL REFERENCE SYSTEM 语句

---

创建或替换空间参照系。

快速链接:

[转至参数](#) (第 52 页)

[转至示例](#) (第 56 页)

[转至用法](#) (第 56 页)

[转至标准](#) (第 57 页)

[转至权限](#) (第 57 页)

#### 语法

```
{ CREATE [ OR REPLACE ] SPATIAL REFERENCE SYSTEM
  | CREATE SPATIAL REFERENCE SYSTEM IF NOT EXISTS }
  srs-name
  [ srs-attribute ] [ srs-attribute ... ]

srs-attribute - (back to Syntax)
  SRID srs-id
  | DEFINITION { definition-string | NULL }
  | ORGANIZATION { organization-name IDENTIFIED BY organization-srs-id
  | NULL }
  | TRANSFORM DEFINITION { transform-definition-string | NULL }
  | LINEAR UNIT OF MEASURE linear-unit-name
  | ANGULAR UNIT OF MEASURE { angular-unit-name | NULL }
  | TYPE { ROUND EARTH | PLANAR }
  | COORDINATE coordinate-name { UNBOUNDED | BETWEEN low-number
AND high-number }
  | ELLIPSOID SEMI MAJOR AXIS semi-major-axis-length { SEMI MINOR AXIS
semi-minor-axis-length
  | INVERSE FLATTENING inverse-flattening-ratio }
  | TOLERANCE { tolerance-distance | DEFAULT }
  | SNAP TO GRID { grid-size | DEFAULT }
  | AXIS ORDER axis-order
  | POLYGON FORMAT polygon-format
  | STORAGE FORMAT storage-format

grid-size - (back to srs-attribute)
  DOUBLE : usually between 0 and 1

axis-order - (back to srs-attribute)
```

```

    { 'x/y/z/m' | 'long/lat/z/m' | 'lat/long/z/m' }

polygon-format - (back to srs-attribute)
    { 'CounterClockWise' | 'Clockwise' | 'EvenOdd' }

storage-format - (back to srs-attribute)
    { 'Internal' | 'Original' | 'Mixed' }

```

## 参数

(返回顶部) (第 51 页)

- **OR REPLACE** - 通过指定 OR REPLACE, 可在数据库中尚未存在空间参照系时创建空间参照系, 而在数据库中已存在空间参照系时替换空间参照系。如果尝试替换正在使用的空间参照系, 则会返回错误。如果在未指定 OR REPLACE 子句的情况下尝试替换已存在于数据库中的空间参照系, 也会返回错误。
- **IF NOT EXISTS** - 指定 CREATE SPATIAL REFERENCE IF NOT EXISTS 检查, 以查看是否已存在使用该名称的空间参照系。如果不存在, 数据库服务器会创建空间参照系。如果确实存在, 则不会继续执行任何操作, 也不会返回错误。
- **IDENTIFIED BY** - 空间参照系的 SRID (*srs-id*)。如果空间参照系由具有 *organization-srs-id* 的组织定义, 则应该将 *srs-id* 设置为该值。

如果未指定 IDENTIFIED BY 子句, 则 SRID 缺省为 ORGANIZATION 子句或 DEFINITION 子句定义的 *organization-srs-id*。如果两个子句都没有定义可以用作缺省 SRID 的 *organization-srs-id*, 则会返回错误。

当空间参照系基于标准坐标系但有不同的测量单位时, 可以将 *srs-id* 值设置成 1000000000 (十亿) 加上标准值。例如, 测量空间参照系 WGS 84 (ID 4326) 的平面测量单位的 SRID 是 1000004326。

除 SRID 0 以外, 由 SAP Sybase IQ 提供的非基于标准系统的空间参照系将指定 2000000000 (二十亿) 或更大的 SRID。从 2000000000 到 2147483647 的 SRID 值范围由 SAP Sybase IQ 保留, 您不应创建该范围内的 SRID。

为减少选择到已由权威定义机构 (例如 OGC) 或其他供应商保留的 SRID 的可能性, 不应选择范围 0 - 32767 (由 EPSG 保留) 或范围 2147483547 - 2147483647 之间的 SRID。

另外, 由于 SRID 作为有符号 32 位整数存储, 所以数值不能超过 231-1, 即 2147483647。

- **DEFINITION** - 设置或替换缺省坐标系设置。如果在 DEFINITION 子句以外的其它子句中设置了任何属性, 则会使用其它子句中指定的值而忽略 DEFINITION 子句中指定的值。

*definition-string* 是由 SQL/MM 和 OGC 定义的空间参考系标准文本语法中的字符串。例如, 以下查询会返回 WGS 84 的定义。

```
SELECT ST_SpatialRefSys::ST_FormatWKT( definition )
FROM ST_SPATIAL_REFERENCE_SYSTEMS
WHERE srs_id=4326;
```

在 Interactive SQL 中，如果双击返回值，则会显示该值更加易读的版本。

当指定 DEFINITION 子句时，会解析 definition-string 并使用它来选择属性的缺省值。例如，definition-string 可能包含 AUTHORITY 元素，该元素定义 organization-name 和 organization-srs-id。

definition-string 中的参数值由使用 SQL 语句子句显式设置的值替代。例如，如果指定 ORGANIZATION 子句，则它将替代 definition-string 中 ORGANIZATION 的值。

- **ORGANIZATION** - 有关创建此空间参照系所基于的空间参照系的组织的信息。
- **TRANSFORM DEFINITION** - 用于转换空间参照系的说明。目前仅支持 PROJ.4 转换。在空间参照系之间转换数据时，由 ST\_Transform 方法使用转换定义。即使未定义 transform-definition-string，仍可使用某些转换。
- **LINEAR UNIT OF MEASURE** - 空间参照系的线性测量单位。指定的值必须与在 ST\_UNITS\_OF\_MEASURE 系统视图中定义的线性测量单位相匹配。

如果未指定该子句，并且没有在 DEFINITION 子句中定义，则缺省值为 METRE。要将预定义测量单位添加到数据库，可以使用 sa\_install\_feature 系统过程。

要将自定义测量单位添加到数据库，可以使用 CREATE SPATIAL UNIT OF MEASURE 语句。

---

**注意：** 虽然可以接受 METRE 和 METER 两种拼写方式，但首选符合 SQL/MM 标准的 METRE。

---

- **ANGULAR UNIT OF MEASURE** - 空间参照系使用的角度测量单位。指定的值必须与在 ST\_UNITS\_OF\_MEASURE 系统表中定义的角度测量单位相匹配。

如果未指定此子句，并且没有在 DEFINITION 子句中定义，则缺省值对于地理空间参照系为 DEGREE，而对于非地理空间参照系为 NULL。

地理空间参照系的角度测量单位必须是非 NULL 值，而非地理空间参照系的角度测量单位必须是 NULL。

地理空间参照系的角度测量单位必须是非 NULL 值，而非地理空间参照系的角度测量单位必须是 NULL。要将预定义测量单位添加到数据库，可以使用 sa\_install\_feature 系统过程。

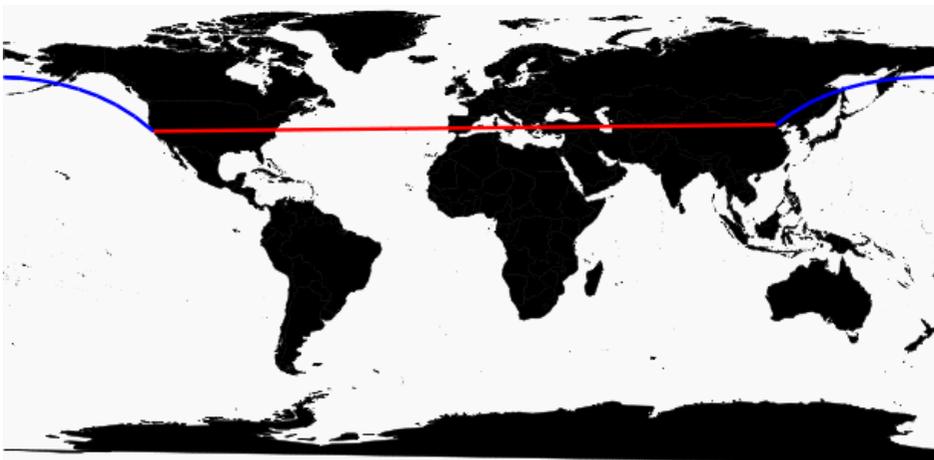
要将自定义测量单位添加到数据库，可以使用 CREATE SPATIAL UNIT OF MEASURE 语句。

- **TYPE** - 控制 SRS 如何解释两点之间的线。对于地理空间参照系，TYPE 子句可以指定 ROUND EARTH (缺省值) 或 PLANAR。ROUND EARTH 模型将两点之间的线解释为大圆弧。给定地球表面上的两个点，则可以选择一个穿过这两点及地心的平面。该平面与地球相交，并且两点之间的线沿此贯穿面是最近的距离。

对于彼此直接相对的两个点，不存在与这两点及地心均相交的唯一平面。连接这些对映点的线段无效且在 **ROUND EARTH** 模型中会报错。

**ROUND EARTH** 模型将地球视为椭球体，并且选择符合地球曲率的线。在某些情况下，如果两点之间的线解释为  $x=\text{long}$ 、 $y=\text{lat}$  的等矩形投影，可能需要使用平面模型。

在下面的示例中，蓝色线显示 **ROUND EARTH** 模型中使用的线解释，而红色线显示对应应在 **PLANAR** 模型中使用的线解释。



**PLANAR** 模型可能用于匹配其它产品使用的解释。由于 **ROUND EARTH** 模型中存在不支持的方法（例如 **ST\_Area**、**ST\_ConvexHull**）和某些部分支持的方法（只有点几何之间支持 **ST\_Distance** 方法）的限制，所以 **PLANAR** 模型仍然非常有用。**ROUND EARTH** 空间参照系不支持基于圆环线串的几何。

对于非地理 **SRS**，类型必须是 **PLANAR**（并且如果没有指定 **TYPE** 子句且未指定 **DEFINITION** 子句或该子句使用非地理定义，则此类型为缺省类型）。

- **COORDINATE** - 空间参照系的维度的范围。coordinate-name 是空间参照系使用的坐标系的名称。对于非地理坐标系，coordinate-name 可以是  $x$ 、 $y$  或  $m$ 。对于地理坐标系，coordinate-name 可以是 **LATITUDE**、**LONGITUDE**、 $z$  或  $m$ 。

指定 **UNBOUNDED** 不设置维度的界限。使用 **BETWEEN** 子句设置上下界限。

**X** 和 **Y** 坐标必须有相关的界限。对于地理空间参照系，缺省情况下经度坐标的范围在  $-180$  到  $180$  度之间，纬度坐标的范围在  $-90$  到  $90$  度之间，除非使用 **COORDINATE** 子句替代这些设置。对于非地理空间参照系，**CREATE** 语句必须同时指定 **X** 和 **Y** 坐标的范围。

**LATITUDE** 和 **LONGITUDE** 用于地理坐标系统。如果没有指定，则 **LATITUDE** 和 **LONGITUDE** 的范围缺省为整个地球。

- **ELLIPSOID** - 在类型为 ROUND EARTH 的空间参照系中将地球表示为椭圆柱体时使用的值。如果 DEFINITION 子句存在，该子句可以指定椭圆柱体定义。如果指定 ELLIPSOID 子句，则它会替换此缺省椭圆柱体。

地球不是正球体，因为地球旋转造成展平而使地心到南北极的距离小于地心到赤道的距离。因此，地球建模为半长轴（地心到赤道的距离）值与半短轴（地心到南北极的距离）值不同的椭圆柱体。虽然使用半长轴与反转展平的方式定义椭圆柱体更为常见，但也可以使用半短轴指定的方法代替（例如，当使用正圆柱体近似地球时必须使用该方法）。半长轴和半短轴通过空间参照系的线性测量单位定义，且反转展平 (1/f) 是比值：

$$1/f = (\text{semi-major-axis}) / (\text{semi-major-axis} - \text{semi-minor-axis})$$

当计算地理空间参照系中的距离时，`product-name` 使用椭圆柱体定义。

- **SNAP TO GRID** - 平面地球（平面）空间参照系，使用 SNAP TO GRID 子句定义执行计算时 SAP Sybase IQ 使用的网格大小。缺省情况下，SAP Sybase IQ 会选择一个网格大小，以便可以在 X 和 Y 边界内的所有点存储 12 位有效数字。例如，如果空间参照系限定 X 在 -180 到 180 之间而 Y 在 -90 到 90 之间，则选择大小为 0.00000001 (1E-9) 的网格。
- **TOLERANCE** - 平面地球（平面）空间参照系，使用 TOLERANCE 子句指定比较点时使用的精度。如果两点之间的距离小于 `tolerance-distance`，则这两点视为相等。设置 `tolerance-distance` 可控制在输入数据中或有限制的内部精度中不精确程度的公差。缺省情况下，`tolerance-distance` 设置为等于 `grid-size`。

设置为 0 时，两点必须完全相等才会视为相等。

对于球形地球空间参照系，TOLERANCE 必须设置为 0。

- **POLYGON FORMAT** - 在内部，SAP Sybase IQ 依据几何组分环的方向解释多边形。当按照定义的点顺序经过圆环时，多边形的内侧在环的左侧。PLANAR 和 ROUND EARTH 空间参照系也应用相同的规则。

SAP Sybase IQ 使用的解释是常见但不通用的解释。有些产品使用完全相反的方向，而有些产品则不依赖环的方向解释多边形。POLYGON FORMAT 子句可用于根据需要进行匹配输入数据的多边形解释。支持以下值：

- **CounterClockwise** - 输入符合 SAP Sybase IQ 的内部解释：根据圆环方向，多边形的内侧在左侧。
- **Clockwise** - 输入采用与 SAP Sybase IQ 解释相反的方法：根据圆环方向，多边形的内侧在右侧。
- **EvenOdd** - （缺省）忽略环方向，而是依据环的嵌套来确定多边形的内侧（外环是最大的环，内环较小，位于外环内部）。从位于环内部的一点引出一条射线，向外辐射穿过所有环。如果穿过的环数是偶数，则为外环。如果是奇数，则为内环。
- **STORAGE FORMAT** - 控制将空间数据加载到数据库时所存储的内容。可能的值为：

- **Internal** – SAP Sybase IQ 仅存储规范化表示。不需要再现原始输入特性时指定此值。这是平面空间参照系 (TYPE PLANAR) 的缺省值。
- **Original** – SAP Sybase IQ 仅存储原始表示。可以再现原始输入特性，但对存储值执行的所有操作必须重复规范化步骤，这可能减慢数据的操作速度。
- **Mixed** – SAP Sybase IQ 可存储内部版本。如果内部版本与原始版本不同，则 SAP Sybase SQL Anywhere® 也会存储原始版本。由于同时存储这两个版本，可以再现原始表示特性，而且对存储值执行的操作不需要重复规范化步骤。但是，由于每个几何都可能存储两个表示，所以存储需求可能会显著增加。**Mixed** 是球形地球空间参照系 (TYPE ROUND EARTH) 的缺省格式。

## 示例

(返回顶部) (第 51 页)

- **示例 1** – 创建名为 mySpatialRS 的空间参照系:

```
CREATE SPATIAL REFERENCE SYSTEM "mySpatialRS"
IDENTIFIED BY 1000026980
LINEAR UNIT OF MEASURE "metre"
TYPE PLANAR
COORDINATE X BETWEEN 171266.736269555 AND 831044.757769222
COORDINATE Y BETWEEN 524881.608973277 AND 691571.125115319
DEFINITION 'PROJCS["NAD83 / Kentucky South",
GEOGCS["NAD83",
DATUM["North American_Datum_1983",
SPHEROID["GRS 1980",
6378137,298.257222101,AUTHORITY["EPSG","7019"]],
AUTHORITY["EPSG","6269"]],
PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]],
UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],
AUTHORITY["EPSG","4269"]],
UNIT["metre",1,AUTHORITY["EPSG","9001"]],
PROJECTION["Lambert_Conformal_Conic_2SP"],
PARAMETER["standard_parallel_1",37.93333333333333],
PARAMETER["standard_parallel_2",36.73333333333333],
PARAMETER["latitude_of_origin",36.33333333333334],
PARAMETER["central_meridian",-85.75],
PARAMETER["false_easting",500000],
PARAMETER["false_northing",500000],
AUTHORITY["EPSG","26980"],
AXIS["X",EAST],
AXIS["Y",NORTH]]'
TRANSFORM DEFINITION '+proj=lcc
+lat_1=37.93333333333333+lat_2=36.73333333333333+lat_0=36.3333333
3333334+lon_0=-85.75+x_0=500000+y_0=500000+ellps=GRS80+datum=NAD8
3+units=m+nö_defs';
```

## 用法

(返回顶部) (第 51 页)

对于地理空间参照系，可以同时指定 **LINEAR** 和 **ANGULAR** 测量单位；而对于非地理空间参照系，只能指定 **LINEAR** 测量单位。**LINEAR** 测量单位用于计算点和区域

之间的距离。**ANGULAR** 测量单位表明如何解释角度经度/纬度，而其对于投影坐标系为 **NULL**，地理坐标系为非 **NULL**。

操作返回的所有派生几何都已经过规划化。

当使用与非 **SQL Anywhere** 数据库同步的数据时，**STORAGE FORMAT** 应设置为 'Original' 或 'Mixed'，以便保留数据的原始特性。

## 标准

(返回顶部) (第 51 页)

ANSI SQL - 遵从性级别: Transact-SQL® 扩充。

## 权限

(返回顶部) (第 51 页)

需要以下特权之一:

- **MANAGE ANY SPATIAL OBJECT** 系统特权。
- **CREATE ANY OBJECT** 系统特权。

# CREATE SPATIAL UNIT OF MEASURE 语句

---

创建或替换空间测量单位。

快速链接:

[转至参数](#) (第 57 页)

[转至示例](#) (第 58 页)

[转至用法](#) (第 58 页)

[转至标准](#) (第 58 页)

[转至权限](#) (第 58 页)

## 语法

```
CREATE [ OR REPLACE ] SPATIAL UNIT OF MEASURE identifier
  TYPE { LINEAR | ANGULAR }
  [ CONVERT USING number ]
```

## 参数

(返回顶部) (第 57 页)

- **OR REPLACE** - 包括 **OR REPLACE** 将创建新的空间测量单位，或替换现有的同名空间测量单位。此子句保留现有特权。如果尝试替换已使用的空间测量单位，则将返回错误。
- **TYPE** - 定义测量单位是用于角度 (**ANGULAR**) 还是用于距离 (**LINEAR**)。
- **CONVERT USING** - 空间单位相对于基本单位的转换系数。对于线性单位，基本单位是 **METRE**。对于角度单位，基本单位是 **RADIAN**。

## 示例

(返回顶部) (第 57 页)

- **示例 1** - 创建名为 **Test** 的空间测量单位:

```
CREATE SPATIAL UNIT OF MEASURE Test
TYPE LINEAR
CONVERT USING 15;
```

## 用法

(返回顶部) (第 57 页)

**CONVERT USING** 子句用于定义如何将定义的测量单位表示的测量值换算为以基本测量单位 (弧度或米) 表示的测量值。用测量值乘以提供的转换系数，就可以算出以基本测量单位表示的测量值。例如，用测量值 512 毫米乘以转换系数 0.001，即可得出测量值 0.512 米。

空间参照系始终包括计算距离 (**ST\_Distance** 或 **ST\_Length**) 或面积时使用的线性测量单位。例如，如果空间参照系的线性测量单位是英里，则使用的面积单位是平方英里。在某些情况下，空间方法接受用于指定要使用的线性测量单位的可选参数。例如，如果空间参照系的线性测量单位是英里，通过使用可选参数 'metre'，可以检索以米表示的两个几何之间的距离。

对于投影坐标系，使用空间参照系的线性单位指定 **X** 和 **Y** 坐标。对于地理坐标系，使用与空间参照系关联的角度测量单位指定经度和纬度。在多数情况下，此角度测量单位是度，但可使用任何有效的角度测量单位。

可以使用 **sa\_install\_feature** 系统过程将预定义的测量单位添加到数据库中。

## 标准

(返回顶部) (第 57 页)

ANSI SQL - 遵从性级别: Transact-SQL 扩充。

## 权限

(返回顶部) (第 57 页)

需要以下特权之一:

- `MANAGE ANY SPATIAL OBJECT` 系统特权。
- `CREATE ANY OBJECT` 系统特权。

## DROP SPATIAL UNIT OF MEASURE 语句

---

删除空间测量单位。

快速链接:

[转至参数](#) (第 59 页)

[转至示例](#) (第 59 页)

[转至标准](#) (第 59 页)

[转至权限](#) (第 59 页)

### 语法

**DROP SPATIAL UNIT OF MEASURE** [ **IF EXISTS** ] *identifier*

### 参数

[\(返回顶部\)](#) (第 59 页)

- **IF EXISTS** - 防止当 **DROP SPATIAL UNIT OF MEASURE** 语句尝试删除不存在的空间测量单位时返回错误。

### 示例

[\(返回顶部\)](#) (第 59 页)

- **示例** - 以下示例删除名为 `Test` 的虚拟空间测量单位:

```
DROP SPATIAL UNIT OF MEASURE Test;
```

### 标准

[\(返回顶部\)](#) (第 59 页)

ANSI SQL - 遵从性级别: Transact-SQL 扩充。

### 权限

[\(返回顶部\)](#) (第 59 页)

需要以下特权之一:

- `MANAGE ANY SPATIAL OBJECT` 系统特权。
- `DROP ANY OBJECT` 系统特权。

- 您拥有空间测量单位。

## DROP SPATIAL REFERENCE SYSTEM 语句

---

删除空间参照系。

快速链接:

[转至参数](#) (第 60 页)

[转至标准](#) (第 60 页)

[转至权限](#) (第 60 页)

### 语法

```
DROP SPATIAL REFERENCE SYSTEM [ IF EXISTS ] name
```

### 参数

([返回顶部](#)) (第 60 页)

- **IF EXISTS** – 防止当 **DROP SPATIAL REFERENCE SYSTEM** 语句尝试删除不存在的空间参照系时返回错误。

### 标准

([返回顶部](#)) (第 60 页)

ANSI SQL - 遵从性级别: Transact-SQL 扩充。

### 权限

([返回顶部](#)) (第 60 页)

需要以下特权之一:

- **MANAGE ANY SPATIAL OBJECT** 系统特权。
- **DROP ANY OBJECT** 系统特权。
- 拥有空间参照系。

## ALTER SPATIAL REFERENCE SYSTEM 语句

---

更改现有空间参照系的设置。

快速链接:

[转至参数](#) (第 61 页)

[转至示例](#) (第 65 页)

转至用法 (第 65 页)

转至标准 (第 65 页)

转至权限 (第 65 页)

## 语法

### ALTER SPATIAL REFERENCE SYSTEM

```
srs-name
[ srs-attribute [ srs-attribute ... ] ]
```

**srs-attribute** - (back to Syntax)

```
SRID srs-id
| DEFINITION { definition-string | NULL }
| ORGANIZATION { organization-name IDENTIFIED BY organization-srs-id
| NULL }
| TRANSFORM DEFINITION { transform-definition-string | NULL }
| LINEAR UNIT OF MEASURE linear-unit-name
| ANGULAR UNIT OF MEASURE { angular-unit-name | NULL }
| TYPE { ROUND EARTH | PLANAR }
| COORDINATE coordinate-name { UNBOUNDED | BETWEEN low-number
AND high-number }
| ELLIPSOID SEMI MAJOR AXIS semi-major-axis-length { SEMI MINOR AXIS
semi-minor-axis-length
| INVERSE FLATTENING inverse-flattening-ratio }
| TOLERANCE { tolerance-distance | DEFAULT }
| SNAP TO GRID { grid-size | DEFAULT }
| AXIS ORDER axis-order
| POLYGON FORMAT polygon-format
| STORAGE FORMAT storage-format
```

**grid-size** - (back to srs-attribute)

**DOUBLE** : usually between 0 and 1

**axis-order** - (back to srs-attribute)

{ '*x/y/z/m*' | '*long/lat/z/m*' | '*lat/long/z/m*' }

**polygon-format** - (back to srs-attribute)

{ '**CounterClockWise**' | '**Clockwise**' | '**EvenOdd**' }

**storage-format** - (back to srs-attribute)

{ '**Internal**' | '**Original**' | '**Mixed**' }

## 参数

(返回顶部) (第 60 页)

- **IDENTIFIED BY** - 空间参照系的 SRID 编号。
- **DEFINITION** - 设置或替换缺省坐标系设置。如果在 **DEFINITION** 子句以外的其它子句中设置了任何属性，则会使用其它子句中指定的值而忽略 **DEFINITION** 子句中指定的值。

*definition-string* 是由 SQL/MM 和 OGC 定义的空间参考系标准文本语法中的字符串。例如，以下查询会返回 WGS 84 的定义。

```
SELECT ST_SpatialRefSys::ST_FormatWKT( definition )
FROM ST_SPATIAL_REFERENCE_SYSTEMS
WHERE srs_id=4326;
```

在 Interactive SQL 中，如果双击返回值，则会显示该值更加易读的版本。

当指定 DEFINITION 子句时，会解析 *definition-string* 并使用它来选择属性的缺省值。例如，*definition-string* 可能包含 AUTHORITY 元素，该元素定义 *organization-name* 和 *organization-srs-id*。

*definition-string* 中的参数值由使用 SQL 语句子句显式设置的值替代。例如，如果指定 ORGANIZATION 子句，则它将替代 *definition-string* 中 ORGANIZATION 的值。

- **ORGANIZATION** - 有关创建此空间参照系所基于的空间参照系的组织的信息。
- **TRANSFORM DEFINITION** - 用于转换空间参照系的说明。目前仅支持 PROJ.4 转换。在空间参照系之间转换数据时，由 ST\_Transform 方法使用转换定义。即使未定义 *transform-definition-string*，仍可使用某些转换。
- **LINEAR UNIT OF MEASURE** - 空间参照系的线性测量单位。指定的值必须与在 ST\_UNITS\_OF\_MEASURE 系统视图中定义的线性测量单位相匹配。

如果未指定该子句，并且没有在 DEFINITION 子句中定义，则缺省值为 METRE。要将预定义测量单位添加到数据库，可以使用 sa\_install\_feature 系统过程。

要将自定义测量单位添加到数据库，可以使用 CREATE SPATIAL UNIT OF MEASURE 语句。

---

**注意：** 虽然可以接受 METRE 和 METER 两种拼写方式，但首选符合 SQL/MM 标准的 METRE。

---

- **ANGULAR UNIT OF MEASURE** - 空间参照系使用的角度测量单位。指定的值必须与在 ST\_UNITS\_OF\_MEASURE 系统表中定义的角度测量单位相匹配。

如果未指定此子句，并且没有在 DEFINITION 子句中定义，则缺省值对于地理空间参照系为 DEGREE，而对于非地理空间参照系为 NULL。

地理空间参照系的角度测量单位必须是非 NULL 值，而非地理空间参照系的角度测量单位必须是 NULL。

地理空间参照系的角度测量单位必须是非 NULL 值，而非地理空间参照系的角度测量单位必须是 NULL。要将预定义测量单位添加到数据库，可以使用 sa\_install\_feature 系统过程。

要将自定义测量单位添加到数据库，可以使用 CREATE SPATIAL UNIT OF MEASURE 语句。

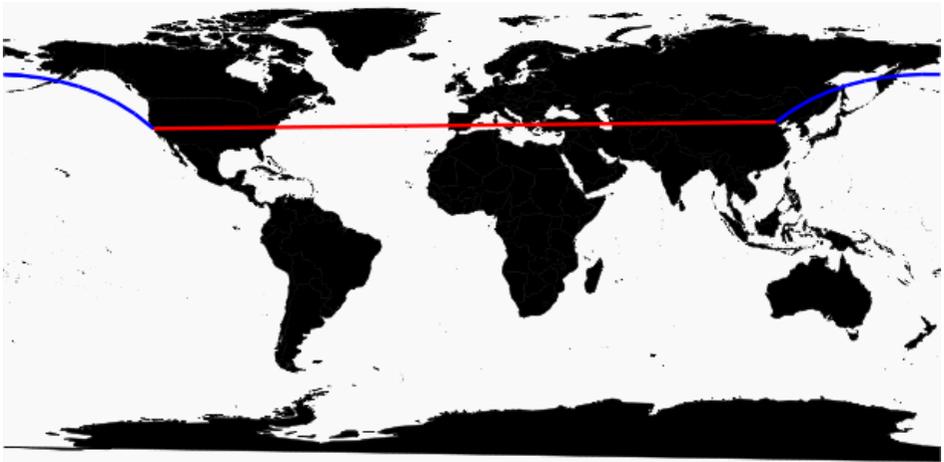
- **TYPE** - 控制 SRS 如何解释两点之间的线。对于地理空间参照系，TYPE 子句可以指定 ROUND EARTH (缺省值) 或 PLANAR。ROUND EARTH 模型将两点之

间的线解释为大圆弧。给定地球表面上的两个点，则可以选择一个穿过这两点及地心的平面。该平面与地球相交，并且两点之间的线沿此贯穿面是最近的距离。

对于彼此直接相对的两个点，不存在与这两点及地心均相交的唯一平面。连接这些对映点的线段无效且在 **ROUND EARTH** 模型中会报错。

**ROUND EARTH** 模型将地球视为椭球体，并且选择符合地球曲率的线。在某些情况下，如果两点之间的线解释为  $x=\text{long}$ 、 $y=\text{lat}$  的等矩形投影，可能需要使用平面模型。

在下面的示例中，蓝色线显示 **ROUND EARTH** 模型中使用的线解释，而红色线显示对应应在 **PLANAR** 模型中使用的线解释。



**PLANAR** 模型可能用于匹配其它产品使用的解释。由于 **ROUND EARTH** 模型中存在不支持的方法（例如 **ST\_Area**、**ST\_ConvexHull**）和某些部分支持的方法（只有点几何之间支持 **ST\_Distance** 方法）的限制，所以 **PLANAR** 模型仍然非常有用。**ROUND EARTH** 空间参照系不支持基于圆弧线串的几何。

对于非地理 **SRS**，类型必须是 **PLANAR**（并且如果没有指定 **TYPE** 子句且未指定 **DEFINITION** 子句或该子句使用非地理定义，则此类型为缺省类型）。

- **COORDINATE** - 空间参照系的维度的范围。**coordinate-name** 是空间参照系使用的坐标系的名称。对于非地理坐标系，**coordinate-name** 可以是 **x**、**y** 或 **m**。对于地理坐标系，**coordinate-name** 可以是 **LATITUDE**、**LONGITUDE**、**z** 或 **m**。

指定 **UNBOUNDED** 不设置维度的界限。使用 **BETWEEN** 子句设置上下界限。

**X** 和 **Y** 坐标必须有相关的界限。对于地理空间参照系，缺省情况下经度坐标的范围在 -180 到 180 度之间，纬度坐标的范围在 -90 到 90 度之间，除非使用 **COORDINATE** 子句替代这些设置。对于非地理空间参照系，**CREATE** 语句必须同时指定 **X** 和 **Y** 坐标的范围。

LATITUDE 和 LONGITUDE 用于地理坐标系统。如果没有指定，则 LATITUDE 和 LONGITUDE 的范围缺省为整个地球。

- **ELLIPSOID** – 在类型为 ROUND EARTH 的空间参照系中将地球表示为椭圆柱体时使用的值。如果 DEFINITION 子句存在，该子句可以指定椭圆柱体定义。如果指定 ELLIPSOID 子句，则它会替换此缺省椭圆柱体。

地球不是正球体，因为地球旋转造成展平而使地心到南北极的距离小于地心到赤道的距离。因此，地球建模为半长轴（地心到赤道的距离）值与半短轴（地心到南北极的距离）值不同的椭圆柱体。虽然使用半长轴与反转展平的方式定义椭圆柱体更为常见，但也可以使用半短轴指定的方法代替（例如，当使用正圆柱体近似地球时必须使用该方法）。半长轴和半短轴通过空间参照系的线性测量单位定义，且反转展平 (1/f) 是比值：

$$1/f = (\text{semi-major-axis}) / (\text{semi-major-axis} - \text{semi-minor-axis})$$

当计算地理空间参照系中的距离时，product-name 使用椭圆柱体定义。

- **SNAP TO GRID** – 平面地球（平面）空间参照系，使用 SNAP TO GRID 子句定义执行计算时 SAP Sybase IQ 使用的网格大小。缺省情况下，SAP Sybase IQ 会选择一个网格大小，以便可以在 X 和 Y 边界内的所有点存储 12 位有效数字。例如，如果空间参照系限定 X 在 -180 到 180 之间而 Y 在 -90 到 90 之间，则选择大小为 0.000000001 (1E-9) 的网格。

- **TOLERANCE** – 平面地球（平面）空间参照系，使用 TOLERANCE 子句指定比较点时使用的精度。如果两点之间的距离小于 tolerance-distance，则这两点视为相等。设置 tolerance-distance 可控制在输入数据中或有限制的内部精度中不精确程度的公差。缺省情况下，tolerance-distance 设置为等于 grid-size。

设置为 0 时，两点必须完全相等才会视为相等。

对于球形地球空间参照系，TOLERANCE 必须设置为 0。

- **POLYGON FORMAT** – 在内部，SAP Sybase IQ 依据几何组分环的方向解释多边形。当按照定义的点顺序经过圆环时，多边形的内侧在环的左侧。PLANAR 和 ROUND EARTH 空间参照系也应用相同的规则。

SAP Sybase IQ 使用的解释是常见但不通用的解释。有些产品使用完全相反的方向，而有些产品则不依赖环的方向解释多边形。POLYGON FORMAT 子句可用于根据需要选择匹配输入数据的多边形解释。支持以下值：

- **CounterClockwise** – 输入符合 SAP Sybase IQ 的内部解释：根据圆环方向，多边形的内侧在左侧。
- **Clockwise** – 输入采用与 SAP Sybase IQ 解释相反的方法：根据圆环方向，多边形的内侧在右侧。
- **EvenOdd** – （缺省）忽略环方向，而是依据环的嵌套来确定多边形的内侧（外环是最大的环，内环较小，位于外环内部）。从位于环内部的一点引出一条射线，向外辐射穿过所有环。如果穿过的环数是偶数，则为外环。如果是奇数，则为内环。

- **STORAGE FORMAT** - 控制将空间数据加载到数据库时所存储的内容。可能的值为：
  - **Internal** - SAP Sybase IQ 仅存储规范化表示。不需要再现原始输入特性时指定此值。这是平面空间参照系 (TYPE PLANAR) 的缺省值。
  - **Original** - SAP Sybase IQ 仅存储原始表示。可以再现原始输入特性，但对存储值执行的所有操作必须重复规范化步骤，这可能减慢数据的操作速度。
  - **Mixed** - SAP Sybase IQ 可存储内部版本。如果内部版本与原始版本不同，则 SAP Sybase SQL Anywhere® 也会存储原始版本。由于同时存储这两个版本，可以再现原始表示特性，而且对存储值执行的操作不需要重复规范化步骤。但是，由于每个几何都可能存储两个表示，所以存储需求可能会显著增加。**Mixed** 是球形地球空间参照系 (TYPE ROUND EARTH) 的缺省格式。

## 示例

(返回顶部) (第 60 页)

- **示例** - 将名为 `mySpatialRef` 的虚构空间参照系的多边形格式更改为 `EvenOdd`：

```
ALTER SPATIAL REFERENCE SYSTEM mySpatialRef
POLYGON FORMAT 'EvenOdd';
```

## 用法

(返回顶部) (第 60 页)

如果有现有数据参照空间参照系，则不能对其进行变更。例如，如果有声明为 `ST_Point(SRID=8743)` 的列，则不能变更 `SRID 8743` 的空间参照系。因为许多空间参照系属性（如存储格式）都会影响数据的存储格式。如果有数据参照 `SRID`，则创建一个新的空间参照系，然后将数据转换为新的 `SRID`。

## 标准

(返回顶部) (第 60 页)

ANSI SQL - 遵从性级别：Transact-SQL 扩充。

## 权限

(返回顶部) (第 60 页)

需要以下特权之一：

- 您是空间参照系的所有者。
- 空间参照系的 `ALTER` 特权。
- `MANAGE ANY SPATIAL OBJECT` 系统特权。
- `ALTER ANY OBJECT` 系统特权。

## ALTER TABLE 语句

---

修改表的定义。

快速链接:

转至参数 (第 68 页)

转至示例 (第 74 页)

转至用法 (第 77 页)

转至标准 (第 77 页)

转至权限 (第 77 页)

### 语法

语法 1 - Alter Owner

```
ALTER TABLE table_name ALTER OWNER TO new_owner
  [ { PRESERVE | DROP } PERMISSIONS ]
  [ { PRESERVE | DROP } FOREIGN KEYS ]
```

语法 2

```
ALTER TABLE [ owner. ] table-name
  [ { ENABLE | DISABLE } RLV STORE
  { alter-clause, ... } ]
```

**alter-clause** - (back to Syntax 2)

**ADD create-clause**

```
| ALTER column-name column-alteration
| ALTER [ CONSTRAINT constraint-name ] CHECK ( condition )
| DROP drop-object
| RENAME rename-object
| move-clause
| SPLIT PARTITION range-partition-name
  INTO ( range-partition-decl, range-partition-decl )
| MERGE PARTITION partition-name-1 INTO partition-name-2
| UNPARTITION
| PARTITION BY
      range-partitioning-scheme
      | hash-partitioning-scheme
      | composite-partitioning-scheme composite-partitioning-scheme
```

**create-clause** - (back to alter-clause)

```
column-name column-definition [ column-constraint ]
| table-constraint
| [ PARTITION BY ] range-partitioning-scheme
```

**column definition** - (back to create-clause)

```
column-name data-type [ NOT NULL | NULL ]
[ IN dbspace-name ]
```

```

[ DEFAULT default-value | IDENTITY ]

column-constraint - (back to create-clause)
[ CONSTRAINT constraint-name ]
{ UNIQUE
| PRIMARY KEY
| REFERENCES table-name [ ( column-name ) ] [ actions ]
| CHECK ( condition )
| IQ UNIQUE ( integer )
}

table-constraint - (back to create-clause)
[ CONSTRAINT constraint-name ]
{ UNIQUE ( column-name [ , ... ] )
| PRIMARY KEY ( column-name [ , ... ] )
| foreign-key-constraint
| CHECK ( condition )
}

foreign-key-constraint - (back to table-constraint)
FOREIGN KEY [ role-name ] [ ( column-name [ , ... ] ) ]
... REFERENCES table-name [ ( column-name [ , ... ] ) ]
... [ actions ]

actions - (back to foreign-key-constraint)
[ ON { UPDATE | DELETE } { RESTRICT } ]

column-alteration - (back to alter-clause)
{ column-data-type | alterable-column-attribute } [ alterable-column-attribute ... ]

| ADD [ constraint-name ] CHECK ( condition )
| DROP { DEFAULT | CHECK | CONSTRAINT constraint-name }

alterable-column-attribute - (back to column-alteration)
[ NOT ] NULL
| DEFAULT default-value
| [ CONSTRAINT constraint-name ] CHECK { NULL | ( condition )
}

default-value - (back to alterable-column-attribute)
CURRENT { DATABASE | DATE | REMOTE USER | TIME | TIMESTAMP | USER |
PUBLISHER )
| string
| global variable
| [ - ] number
| ( constant-expression )
| built-in-function ( constant-expression )
| AUTOINCREMENT
| NULL
| TIMESTAMP
| LAST USER
| USER

drop-object - (back to alter-clause)
{ column-name

```

```

| CHECK constraint-name
| CONSTRAINT
| UNIQUE ( index-columns-list )
| PRIMARY KEY
| FOREIGN KEY fkey-name
| [ PARTITION ] range-partition-name
}

rename-object - (back to alter-clause)
  new-table-name
  | column-name TO new-column-name
  | CONSTRAINT constraint-name TO new-constraint-name
  | [ PARTITION ] range-partition-name TO new-range-partition-name

move-clause - (back to alter-clause)
  { ALTER column-name
    MOVE
    { PARTITION ( range-partition-name TO new-dbspace-name )
      | TO new-dbspace-name }
    }
  | MOVE PARTITION range-partition-name TO new-dbspace-name
  | MOVE TO new-dbspace-name
  | MOVE TABLE METADATA TO new-dbspace-name
  }

```

```

range-partitioning-scheme - (back to alter-clause)
  RANGE ( partition-key )
  ( range-partition-decl [, range-partition-decl ... ] )

partition-key - (back to range-partitioning-scheme)
  column-name

range-partition-decl - (back to alter-clause) or (back to range-
partitioning-scheme)
  range-partition-name VALUES <= ( { constant | MAX } ) [ IN dbspace-
name ]

hash-partitioning-scheme - (back to alter-clause) or (back to composite-
partitioning-scheme)
  HASH ( partition-key, ... ] )

composite-partitioning-scheme - (back to alter-clause)
  hash-partitioning-scheme SUBPARTITION range-partitioning-scheme

```

### 参数

(返回顶部) (第 66 页)

- **{ ENABLE | DISABLE } RLV STORE** - 将该表注册到 RLV 存储库以进行实时内存更新。不支持 IQ 临时表，也不能在 Multiplex 环境中运行。此值替代数据库选项 **BASE\_TABLES\_IN\_RLV** 的值。
- **ADD column-definition [ column-constraint ]** - 向表中添加新列。

表必须为空才可指定 NOT NULL。添加 IDENTITY 或 DEFAULT AUTOINCREMENT 列时，表可能包含数据。如果列有缺省 IDENTITY 值，新列的所有行将用连续值

来填充。也可添加 FOREIGN 约束作为单列键的列约束。IDENTITY/DEFAULT AUTOINCREMENT 列的值唯一标识表中的每一行。

IDENTITY/DEFAULT AUTOINCREMENT 列存储在插入和更新期间自动生成的顺序编号。DEFAULT AUTOINCREMENT 列也称为 IDENTITY 列。使用 IDENTITY/DEFAULT AUTOINCREMENT 时，列必须是小数位数为 0 的整数数据类型之一或精确的数值类型。有关列约束和 IDENTITY/DEFAULT AUTOINCREMENT 列的详细信息，请参见 CREATE TABLE 语句。

**IQ UNIQUE** 约束 - 定义列的预期基数并确定是将该列装载为 Flat FP 还是 NBit FP。IQ UNIQUE(*n*) 值显式设置为 0 会将该列装载为 Flat FP。不带 IQ UNIQUE 约束的列将隐式装载为 NBit，一直到 FP\_NBIT\_AUTOSIZE\_LIMIT、FP\_NBIT\_LOOKUP\_MB 和 FP\_NBIT\_ROLLOVER\_MAX\_MB 选项定义的限制。

不必使用 *n* 值小于 FP\_NBIT\_AUTOSIZE\_LIMIT 的 IQ UNIQUE。自动大小调整功能会自动将所有中低基数列的大小调整为 NBit。如果想要将列装载为 Flat FP，或者想要在不同值的数量超出 FP\_NBIT\_AUTOSIZE\_LIMIT 时将列装载为 NBit，请使用 IQ UNIQUE。

---

#### 注意:

- 当指定高 IQ UNIQUE 值时，请考虑内存的使用情况。如果计算机资源受到限制，应避免带有 FP\_NBIT\_ENFORCE\_LIMITS='OFF' (缺省设置) 的装载。在 SAP Sybase IQ 16.0 之前，IQ UNIQUE *n* 值 > 16777216 时会切换到 Flat FP。在 16.0 中，支持对较大的 IQ UNIQUE 值进行标识化，但鉴于基数和列宽，该值可能需要非常多的内存资源。
- BIT、BLOB 和 CLOB 数据类型不支持 NBit 字典压缩。如果 FP\_NBIT\_IQ15\_COMPATIBILITY='OFF'，则在包含这些数据类型的 CREATE TABLE 或 ALTER TABLE 语句中指定的非零 IQ UNIQUE 列将返回一个错误。

---

#### • ALTER *column-name* column-alteration - 更改列定义:

- **SET DEFAULT** *default-value* - 更改表中现有列的缺省值。您也可以为此任务使用 MODIFY 子句，但 ALTER 与 ISO/ANSI SQL 兼容，而 MODIFY 与其不兼容。修改缺省值不会更改表中的任何现有值。
- **DROP DEFAULT** - 删除表中现有列的缺省值。您也可以为此任务使用 MODIFY 子句，但 ALTER 与 ISO/ANSI SQL 兼容，而 MODIFY 与其不兼容。删除缺省值不会更改表中的任何现有值。
- **ADD** - 在列中添加已命名约束或 CHECK 条件。新约束或条件仅在定义后应用于对表的操作。但不会对表中现有的值进行校验来确认其是否满足新约束或条件。
- **CONSTRAINT** *column-constraint-name* - 可选列约束名用于以后单独修改或删除约束，而不必修改整个列的约束。

- [ **CONSTRAINT** *constraint-name* ] **CHECK** ( *condition* ) – 此子句用于在列上添加 CHECK 约束。
- **SET COMPUTE** ( *expression* ) – 更改与计算列关联的表达式。当执行语句时，重新计算列中的值，如果新表达式无效，则语句执行失败。
- **DROP COMPUTE** – 将列从计算列更改为非计算列。此语句不更改表中的任何现有值。
- **ADD table-constraint** – 向表中添加约束。

也可单列键或多列键添加外键约束作为表约束。如果指定 **PRIMARY KEY**，表不得具有通过 **CREATE TABLE** 语句或其它 **ALTER TABLE** 语句创建的主键。有关表约束的完整说明，请参见 **CREATE TABLE** 语句。

---

**注意：** 不能通过 **MODIFY** 修改表约束或列约束。要更改约束，必须通过 **DELETE** 删除旧约束并通过 **ADD** 添加新约束。

---

- **DROP drop-object** – 删除表对象：
  - **DROP column-name** – 从表中删除列。如果列包含在任何多列索引、唯一性约束、外键或主键中，则必须删除索引、约束或键后才能删除列。这不会删除引用该列的 CHECK 约束。只有在 **IDENTITY\_INSERT** 处于关闭状态并且表不是局部临时表的情况下，才能删除 **IDENTITY/DEFAULT AUTOINCREMENT** 列。
  - **DROP CHECK** – 删除表的所有检查约束。这包括表检查约束和列检查约束。
  - **DROP CONSTRAINT constraint-name** – 删除表或指定列的已命名约束。
  - **DROP UNIQUE** ( *column-name, ...* ) – 删除指定列上的唯一约束。同时也删除引用该唯一约束的任何外键（而不是主键）。如果有关联的外键约束，系统会报告错误。在删除主键约束之前，使用 **ALTER TABLE** 删除所有引用该主键的外键。
  - **DROP PRIMARY KEY** – 删除主键。同时也删除引用此表主键的所有外键。如果有关联的外键约束，系统会报告错误。如果主键是未实施的主键，并且存在关联的未实施的外键约束，则 **DELETE** 会返回错误。
  - **DROP FOREIGN KEY role-name** – 删除该表的具有给定角色名称的外键约束。系统将保留为该外键约束隐式创建的非唯一 **HG** 索引。用户可以使用 **DROP INDEX** 语句显式删除 **HG** 索引。
  - **DROP [ PARTITION ]** – 删除指定分区。分区 **p1** 中的行将被删除，分区定义也会被删除。不能删除最后一个分区，因为删除最后一个分区会将分区表转变为未分区的表。（要合并分区表，请改用 **UNPARTITION** 子句。）例如：

```
CREATE TABLE foo (c1 INT, c2 INT)
PARTITION BY RANGE (c1)
(P1 VALUES <= (100) IN dbsp1,
 P2 VALUES <= (200) IN dbsp2,
 P3 VALUES <= (MAX) IN dbsp3
) IN dbsp4);
LOAD TABLE ...;
ALTER TABLE DROP PARTITION P1;
```

- **RENAME *rename-object*** - 重命名表中的对象：
  - **RENAME *new-table-name*** - 将表名更改为 *new-table-name*。必须修改任何使用旧表名的应用程序。此外，系统自动分配了旧表名的任何外键都不会更改名称。
  - **RENAME *column-name* TO *new-column-name*** - 将列名更改为 *new-column-name*。必须修改任何使用旧列名的应用程序。
  - **RENAME [ PARTITION ]** - 重命名现有分区。
  - **RENAME *constraint-name* TO *new-constraint-name*** - 将约束名更改为 *new-constraint-name*。必须修改任何使用旧约束名的应用程序。
- **MOVE clause** - 移动表对象。表对象只能驻留在一个 **dbspace** 内。任何类型的 ALTER MOVE 都会阻止在整个移动期间内对表执行任何修改。

---

**注意：** 无法将对象移动到高速缓存 **dbspace**。

---

- **MOVE TO** - 当表映射至新的 **dbspace** 时，移动同一 **dbspace** 内驻留的所有表对象，其中包括列、索引、唯一约束、主键、外键和元数据。不能在分区表上请求 ALTER Column MOVE TO 子句。  
无法将 BIT 数据类型列显式置于 **dbspace** 中。BIT 数据类型不支持以下内容：  

```
ALTER TABLE t2 alter c1_bit MOVE TO iq_main;
```
- **MOVE TABLE METADATA** - 将表的元数据移动到新的 **dbspace**。对于分区表，MOVE TABLE METADATA 还会移动各分区之间共享的元数据。
- **MOVE PARTITION** - 将指定分区移动到新的 **dbspace**。
- **PARTITION BY** - 将大表拆分为多个便于管理的小存储对象。各个分区共用父表的相同逻辑属性，但可放置在单独的 **dbspace** 中并分别进行管理。SAP Sybase IQ 支持多个表分区模式：
  - 散列分区
  - 范围分区
  - 复合分区

分区键是包含表分区键的一个或多个列。分区键可包含 NULL 和 DEFAULT 值，但不能包含：

- LOB (BLOB 或 CLOB) 列
- BINARY 或 VARBINARY 列
- 长度超过 255 个字节的 CHAR 或 VARCHAR 列
- BIT 列
- FLOAT/DOUBLE/REAL 列
- **PARTITION BY RANGE** - 根据分区列中的值范围对行进行分区。范围分区被限定为有 1 个分区键列和最多 1024 个分区。在范围分区模式中，分区键是包含表分区键的列：

```
range-partition-decl:
  partition-name VALUES <= ( {constant-expr | MAX } [ ,
```

```
{ constant-expr | MAX } ]... )
  [ IN dbspace-name ]
```

分区名称是存储表行的新分区的名称。在表分区集合中，分区名称必须唯一。分区名称是必需项。

- **VALUE** - 按升序为每个分区指定上限（含上限）。用户必须为每个范围分区指定分区条件，确保将每一行只分配到一个分区。分区列允许 NULL 值，将 NULL 作为分区键值的行属于第一个表分区。但 NULL 不能是界限值。

第一个分区没有下限（MIN 值）。分区键第一列中单元格值为 NULL 的行属于第一个分区。对于最后一个分区，您可以指定一个上限（含上限）或 MAX。如果最后一个分区上限值不是 MAX，则加载或插入其分区键值大于最后一个分区上限值的任何行都将生成错误。

- **Max** - 表示无限大的上限，只能为最后一个分区指定。
- **IN** - 在 partition-decl 中指定其中应该包含分区行的 dbspace。

以下限制影响范围分区表的分区键和界限值：

- 只有所有现有行都属于第一分区，才能对未分区表进行范围分区。
- 分区界限必须是常量，而不是常量表表达式。
- 分区界限必须按照分区的创建顺序以升序排列。也就是说，第二个分区上限必须大于第一个分区上限，依此类推。

此外，分区界限值必须与相应的分区键列数据类型兼容。例如，VARCHAR 与 CHAR 兼容。

- 如果某个界限值的数据类型不同于其对应分区键列的数据类型，SAP Sybase IQ 会将界限值的数据类型转换为分区键列的数据类型，但存在以下例外情况：
- 不允许进行显式转换。以下示例尝试将 INT 显式转换为 VARCHAR，并生成错误：

```
CREATE TABLE Employees(emp_name VARCHAR(20))
PARTITION BY RANGE(emp_name)
(p1 VALUES <=(CAST (1 AS VARCHAR(20))),
p2 VALUES <=(CAST (10 AS VARCHAR(20)))
```

- 不允许执行可导致数据丢失的隐式转换。在以下示例中，分区界限与分区键数据类型不兼容。舍入假设可导致数据丢失，并生成错误：

```
CREATE TABLE emp_id (id INT) PARTITION BY RANGE(id) (p1 VALUES
<=(10.5), p2 VALUES <=(100.5))
```

- 在以下示例中，分区界限与分区键数据类型兼容。界限值会直接转换为浮点值。无需进行舍入处理，且支持转换：

```
CREATE TABLE id_emp (id FLOAT)
PARTITION BY RANGE(id) (p1 VALUES <=(10),
p2 VALUES <=(100))
```

- 不允许将非二进制数据类型转换为二进制数据类型。例如，不允许进行以下转换，并会返回错误：

```
CREATE TABLE newemp (name BINARY)
PARTITION BY RANGE(name)
```

```
(p1 VALUES <= ("Maarten"),
p2 VALUES <= ("Zymmerman"))
```

- 在范围分区表中，不能将 **NULL** 用作界限。
- 如果分区键第一列中的单元格值计算结果为 **NULL**，则对应行将属于第一个分区。**SAP Sybase IQ** 仅支持一列分区键，因此分区键中的任何 **NULL** 都会将对应行分配到第一个分区。

- PARTITION BY HASH** - 基于内部散列函数处理的分区-键值向各个分区映射数据。散列分区键被限定为最多有 8 个列，其声明的列宽总和小于等于 5300 个字节。对于散列分区，表创建者仅确定分区键列；分区的数量和位置将在内部确定。

在散列分区声明中，分区键是一个或多个列，其合成值将确定存储有各行数据的分区：

```
hash-partitioning-scheme:
HASH ( partition-key [ , partition-key, ... ] )
```

- 限制 -**

- 只能对基表进行散列分区。试图对全局临时表或局部临时表进行分区将会引发错误。
- 只能对空的未分区表进行散列分区。
- 不能添加、删除、合并或拆分散列分区。
- 不能为散列分区键添加列或从中删除列。

- PARTITION BY HASH RANGE** - 按范围对已进行散列分区的表划分子分区。在散列范围分区模式声明中，**SUBPARTITION BY RANGE** 子句将向现有散列范围分区表添加一个新的范围子分区：

```
hash-range-partitioning-scheme:
PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
  [ SUBPARTITION BY RANGE ( range-partition-decl [ , range-
partition-decl ... ] ) ]
```

散列分区指定如何在逻辑上分布和托管数据；范围子分区指定如何在物理上放置数据。新的范围子分区通过散列进行逻辑分区，其散列分区键与现有的散列范围分区表相同。范围子分区键被限定为仅一列。

- 限制 -**

- 只能对基表进行散列分区。试图对全局临时表或局部临时表进行分区将会引发错误。
- 只有散列分区表为空表时才能按范围对其进行子分区。
- 不能添加、删除、合并或拆分散列分区。
- 不能为散列分区键添加列或从中删除列。

---

**注意：**与散列范围分区相同，范围分区和复合分区模式要求有单独授权的 VLDB 管理组件。

---

- MERGE PARTITION** - 将 *partition-name-1* 合并到 *partition-name-2*。如果两个分区相邻且数据驻留在同一个 **dbspace** 内，则可合并这两个分区。只能将分区值较低

的分区合并到分区值较高的相邻分区内。请注意，服务器不会检查分区所合并到的 `dbspace` 的 `CREATE` 特权。有关如何创建相邻分区的示例，请参见 `CREATE TABLE` 语句示例。

- **RENAME PARTITION** – 重命名现有 `PARTITION`。
- **UNPARTITION** – 从分区表中删除分区。每一列都置于单个 `dbspace` 内。请注意，服务器不会检查所有分区数据将移动到的目标 `dbspace` 的 `CREATE` 特权。`ALTER TABLE UNPARTITION` 会阻塞所有数据库活动。
- **ALTER OWNER** – 更改表的所有者。`ALTER OWNER` 子句不能与 `ALTER TABLE` 语句的任何其它 `[alter-clause]` 子句配合使用。
  - **[ PRESERVE | DROP ] PERMISSIONS** – 如果不希望新的所有者拥有与旧所有者相同的特权，则可使用 `DROP` 特权子句（缺省）来删除允许用户访问该表的所有显式授予的特权。隐式授予该表所有者的特权将被授予新的所有者，并从旧的所有者删除这些特权。
  - **[ PRESERVE | DROP ] FOREIGN KEYS** – 如果想防止新的所有者访问被引用表中的数据，可使用 `DROP FOREIGN KEYS` 子句（缺省）来删除该表中的所有外键以及引用该表的所有外键。除非所有引用表都属于新的所有者，否则无法将 `PRESERVE FOREIGN KEYS` 子句与 `DROP PERMISSIONS` 子句配合使用。

在以下情况下，`ALTER TABLE ALTER OWNER` 语句会失败：

- 存在另一个与原始表同名的表，并且该表属于新用户。
- 同时指定了 `PRESERVE FOREIGN KEYS` 和 `PRESERVE PERMISSIONS` 子句，并且有一个外键所属的用户不是引用该表的新表所有者，而且该外键依赖于隐式授予的特权（如授予表所有者的特权）。为了避免语句失败，可将 `SELECT` 特权显式授予引用表的原始所有者，或者删除外键。
- 指定了 `PRESERVE FOREIGN KEYS` 子句，但未指定 `PRESERVE PERMISSIONS` 子句，并且有一个外键所属的用户不是引用该表的新表所有者。为了避免语句失败，可删除外键。
- 指定了 `PRESERVE FOREIGN KEYS` 子句，并且表中包含一个依赖于隐式授予的特权（如授予表所有者的特权）的外键。为了避免语句失败，可将被引用表的 `SELECT` 特权显式授予新所有者，或者删除外键。
- 表包含一个具有缺省值且引用序列的列，而序列生成器的 `USAGE` 特权依赖于隐式授予的特权（如授予序列所有者的特权）。为了避免语句失败，可将序列生成器的 `USAGE` 特权显式授予表的新所有者。
- 存在依赖于原始表的已启用实例化视图。

## 示例

(返回顶部) (第 66 页)

- **示例 1** – 向 `Employees` 表添加一个新列，说明他们在哪个办公室工作：

```
ALTER TABLE Employees
ADD office CHAR(20)
```

- **示例 2** - 从 Employees 表中删除 office 列:

```
ALTER TABLE Employees
DROP office
```

- **示例 3** - 向 Customers 表添加一列, 用于为每个客户指派一个销售联系人:

```
ALTER TABLE Customers
ADD SalesContact INTEGER
REFERENCES Employees (EmployeeID)
```

- **示例 4** - 向 Customers 表添加一个新列 CustomerNum, 并指定缺省值 88:

```
ALTER TABLE Customers
ADD CustomerNum INTEGER DEFAULT 88
```

- **示例 5** - 将 c2、c4 和 c5 的 **FP** 索引从 dbspace Dsp3 移动到 Dsp6。c1 的 **FP** 索引保留在 Dsp1 中。c3 的 **FP** 索引保留在 Dsp2 中。c5 的主键保留在 Dsp4 中。**DATE** 索引 c4\_date 保留在 Dsp5 中。

```
CREATE TABLE foo (
    c1 INT IN Dsp1,
    c2 VARCHAR(20),
    c3 CLOB IN Dsp2,
    c4 DATE,
    c5 BIGINT,
    PRIMARY KEY (c5) IN Dsp4) IN Dsp3);

CREATE DATE INDEX c4_date ON foo(c4) IN Dsp5;
ALTER TABLE foo
MOVE TO Dsp6;
```

- **示例 6** - 只将 **FP** 索引 c1 从 dbspace Dsp1 移动到 Dsp7:

```
ALTER TABLE foo ALTER c1 MOVE TO Dsp7
```

- **示例 7** - 使用多个 **ALTER TABLE** 子句对分区进行移动、拆分、重命名和合并。

创建分区表:

```
CREATE TABLE bar (
    c1 INT,
    c2 DATE,
    c3 VARCHAR(10))
PARTITION BY RANGE(c2)
(p1 VALUES <= ('2005-12-31') IN dbbsp1,
 p2 VALUES <= ('2006-12-31') IN dbbsp2,
 p3 VALUES <= ('2007-12-31') IN dbbsp3,
 p4 VALUES <= ('2008-12-31') IN dbbsp4);
INSERT INTO bar VALUES(3, '2007-01-01', 'banana nut');
INSERT INTO bar VALUES(4, '2007-09-09', 'grape jam');
INSERT INTO bar VALUES(5, '2008-05-05', 'apple cake');
```

将分区 p2 移动到 dbbsp5:

```
ALTER TABLE bar MOVE PARTITION p2 TO DBSP5;
```

将分区 p4 拆分为 2 个分区:

```
ALTER TABLE bar SPLIT PARTITION p4 INTO
  (P41 VALUES <= ('2008-06-30') IN dbsp4,
   P42 VALUES <= ('2008-12-31') IN dbsp4);
```

下面的 **SPLIT PARTITION** 报告一个错误, 因为它需要移动数据。拆分后, 并非所有现有行都位于同一分区内。

```
ALTER TABLE bar SPLIT PARTITION p3 INTO
  (P31 VALUES <= ('2007-06-30') IN dbsp3,
   P32 VALUES <= ('2007-12-31') IN dbsp3);
```

会报告以下错误:

```
No data move is allowed, cannot split partition p3.
```

下面的 **SPLIT PARTITION** 报告一个错误, 因为它更改了分区的边界值:

```
ALTER TABLE bar SPLIT PARTITION p2 INTO
  (p21 VALUES <= ('2006-06-30') IN dbsp2,
   P22 VALUES <= ('2006-12-01') IN dbsp2);
```

会报告以下错误:

```
Boundary value for the partition p2 cannot be changed.
```

将分区 p3 合并到 p2。由于不允许将边界值较高的分区合并到边界值较低的分  
区, 因此, 系统会报告错误。

```
ALTER TABLE bar MERGE PARTITION p3 into p2;
```

会报告以下错误:

```
Partition 'p2' is not adjacent to or before partition 'p3'.
```

将分区 p2 合并到 p3:

```
ALTER TABLE bar MERGE PARTITION p2 INTO P3;
```

将分区 p1 重命名为 p1\_new:

```
ALTER TABLE bar RENAME PARTITION p1 TO p1_new;
```

对表 bar 取消分区:

```
ALTER TABLE bar UNPARTITION;
```

对表 bar 进行分区。此命令会报告错误, 因为所有行必须位于第一个分区中。

```
ALTER TABLE bar PARTITION BY RANGE (c2)
  (p1 VALUES <= ('2005-12-31') IN dbsp1,
   P2 VALUES <= ('2006-12-31') IN DBSP2,
   P3 VALUES <= ('2007-12-31') IN dbsp3,
   P4 VALUES <= ('2008-12-31') IN dbsp4);
```

会报告以下错误:

```
All rows must be in the first partition.
```

对表 bar 进行分区:

```
ALTER TABLE bar PARTITION BY RANGE (c2)
  (p1 VALUES <= ('2008-12-31') IN dbsp1,
   p2 VALUES <= ('2009-12-31') IN dbsp2,
   p3 VALUES <= ('2010-12-31') IN dbsp3,
   p4 VALUES <= ('2011-12-31') IN dbsp4);
```

- **示例 8** - 更改表 tab1，使其不再注册到 RLV 存储库中以进行实时内存更新。

```
ALTER TABLE tab1 DISABLE RLV STORE
```

## 用法

(返回顶部) (第 66 页)

**ALTER TABLE** 语句可更改以前创建的表中的表属性（列定义和约束）。该语法允许使用一组变更子句；但在每个 **ALTER TABLE** 语句中，只能添加、修改或删除一个表约束或列约束。只要 **ALTER TABLE** 语句影响了当前正由其它连接使用的表，就会禁止该语句。**ALTER TABLE** 可能很耗时，并且在处理语句时，服务器不会处理引用相同表的请求。

---

**注意：** 您不能更改本地临时表，但可以在仅有一个连接使用全局临时表时，更改全局临时表。

---

SAP Sybase IQ 强制执行 **REFERENCES** 和 **CHECK** 约束。只有在满足以下条件时，才会评估添加到 **ALTER TABLE** 语句的表和/或列检查约束：变更表操作过程中在所添加的其中一个新列中定义这些约束。有关 **CHECK** 约束的详细信息，请参见“**CREATE TABLE** 语句”。

如果在视图定义中使用 **SELECT \***，并且变更 **SELECT \*** 所引用的表，则您必须运行 **ALTER VIEW <viewname> RECOMPILE** 以确保视图定义正确，并防止在查询该视图时出现意外的结果。

**副作用：**

- 自动提交。**ALTER** 和 **DROP** 选项用于关闭当前连接的所有游标。Interactive SQL 数据窗口同时也会被清除。
- **ALTER TABLE** 操作开始时执行检查点。
- 一旦更改了列或表，引用更改列的任何存储过程、视图或其它项将不再有效。

## 标准

(返回顶部) (第 66 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 一些子句受 SAP Adaptive Server® Enterprise 的支持。

## 权限

(返回顶部) (第 66 页)

**句法 1**

需要以下特权之一：

- ALTER ANY TABLE 系统特权
- ALTER ANY OBJECT 系统特权
- 表的 ALTER 特权
- 您拥有该表

**句法 2**

语法 1 所需的系统特权因使用的子句而异。

子句	所需特权
Add	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 基础表的 ALTER 特权</li> <li>• 您拥有基础表</li> </ul> <p>UNIQUE、PRIMARY KEY、FOREIGN KEY 或 IQ UNIQUE 列约束 – 需要以上特权，外加基础表的 REFERENCES 特权。</p> <p>FOREIGN KEY 表约束需要以上特权，外加以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY INDEX 系统特权</li> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 基表的 REFERENCES 特权</li> </ul> <p>PARTITION BY RANGE 需要以上特权，外加下列特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 对要创建分区的 dbspace 的 CREATE 特权</li> </ul>
Alter	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> <li>• 您拥有该表。</li> </ul> <p>变更主键或唯一约束还需要表的 REFERENCES 特权。</p>

子句	所需特权
Drop	<p>删除不含约束的列 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY OBJECT 系统特权</li> <li>• ALTER ANY TABLE 系统特权</li> <li>• 基础表的 ALTER 特权</li> <li>• 您拥有基础表</li> </ul> <p>删除含有约束的列或表需要以上特权以及 REFERENCES 特权（如果使用 ALTER 特权）。</p> <p>删除自身拥有的表分区 - 无需任何权限。</p> <p>删除其他用户拥有的表分区 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> </ul>
RENAME	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> <li>• 您拥有该表</li> </ul>
Move	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 系统特权</li> <li>• 基础表的 ALTER 特权</li> <li>• 您拥有基础表</li> </ul> <p>另外，还需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 分区要移动到的目标 dbspace 的 CREATE 特权</li> </ul>

子句	所需特权
Split Partition	<p>将自身拥有的表分区 - 无需任何权限。</p> <p>将其他用户拥有的表分区 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• SELECT ANY TABLE 系统特权</li> <li>• 表的 SELECT 特权</li> </ul> <p>另外，还需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> </ul>
Merge Partition、Unpartition	<p>自身拥有的表 - 无需任何权限。</p> <p>其他用户拥有的表 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> </ul>
Partition By	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 对要创建分区的 dbspace 的 CREATE 特权</li> </ul> <p>另外，还需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> <li>• 您拥有该表</li> </ul>
或 disable RLV store	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> </ul>

# 索引

## A

ALTER TABLE 语句  
语法 66  
ALTER VIEW 语句  
RECOMPILE 66

## B

表  
更改 66  
更改定义 66

## F

分区  
删除 66

## K

空间参照系  
变更 20

创建 20  
删除 20  
空间测量单位  
创建 21  
删除 21

## L

列  
更改 66

## R

REFERENCES 子句 66

## S

SELECT \* 66  
删除分区 66  
视图  
视图中的变更表 66

