



管理：空間データ

---

# SAP Sybase IQ 16.0 SP03

ドキュメント ID：DC02013-01-1603-01

改訂：2013年12月

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG による明示的な許可なく複製または転載することは、形態や目的を問わず禁じられています。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダ独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

# 目次

<b>制限事項</b> .....	<b>1</b>
<b>空間データ</b> .....	<b>3</b>
空間参照系 (SRS) と空間参照系識別子 (SRID) .....	4
測定単位 .....	7
追加の事前定義済み測定単位のインストール .....	7
<b>SAP Sybase IQ</b> による空間データのサポート .....	8
サポートされる空間データ型とその階層 .....	8
空間標準への準拠 .....	11
サポートと準拠についての特記事項 .....	12
サポートされる空間データのインポート フォーマットとエクスポートフォーマット ....	13
<b>ESRI</b> シェイプファイルのサポート .....	15
空間トピック関連の推奨ドキュメント .....	16
空間カラムの作成 (SQL の場合) .....	16
空間カラムのインデックス .....	17
空間データ型の構文 .....	17
ジオメトリの作成方法 .....	20
空間データのイメージとしての表示 (Interactive SQL の場合) .....	21
空間データのイメージとしての表示 (Spatial Viewer の場合) .....	22
<b>WKT (Well Known Text)</b> ファイルからの空間データの ロード .....	23
空間参照系の作成または管理 .....	24
空間測定単位の作成または管理 .....	25
<b>空間に関する高度なトピック</b> .....	<b>27</b>
平面および曲面の表現方法 .....	27

「グリッドにスナップ」と許容度が空間の計算に与える影響 .....	28
補間が空間の計算に与える影響 .....	32
多角形リングの方向操作 .....	33
ジオメトリの内部、外部、境界の操作 .....	34
空間の比較操作 .....	35
空間の関係操作 .....	36
空間の次元の操作 .....	39
<b>チュートリアル：空間機能の実験 .....</b>	<b>41</b>
レッスン 1：追加の測定単位と空間参照系のインストール .....	41
レッスン 2：ESRI シェイプファイルデータのダウンロード .....	42
レッスン 3：ESRI シェイプファイルデータのロード .....	43
レッスン 4：空間データのクエリ .....	46
レッスン 5：SVG への空間データの出力 .....	48
レッスン 6：空間データの投影 .....	50
<b>空間データへのアクセスとそのデータの操作 .....</b>	<b>53</b>
<b>付録：SQL 文 .....</b>	<b>55</b>
CREATE SPATIAL REFERENCE SYSTEM 文 .....	55
CREATE SPATIAL UNIT OF MEASURE 文 .....	63
DROP SPATIAL UNIT OF MEASURE 文 .....	64
DROP SPATIAL REFERENCE SYSTEM 文 .....	65
ALTER SPATIAL REFERENCE SYSTEM 文 .....	66
ALTER TABLE 文 .....	73
<b>索引 .....</b>	<b>91</b>

## 制限事項

SAP® Sybase® IQ で空間データ、空間参照系、および空間測定単位を操作する前に、3D メソッドのパフォーマンスおよび参照整合性に関する制限事項と影響をよく理解してください。

### 2006 ISO 標準

3D メソッドはサポートされていませんが、Z 次元と M 次元を格納できます。2006 ISO 標準では、2D 空間メソッド (X 次元と Y 次元) のみがサポートされています。

### 空間データを IQ カタログストアテーブルに格納する必要がある

空間データ、空間参照系、および空間測定単位は、カタログストアでのみ使用できます。IQ メインストアでは、空間データの解釈や格納はできません。カタログストア内の空間データを問い合わせる IQ メインストアテーブルにジョインできませんが、このジョインは非空間カラムに対して行う必要があります。

たとえば、IQ メインストアテーブルに格納された顧客のそれぞれに対して ST\_Point を関連付ける必要があるとします。IQ メインストアテーブルには ST\_Point を格納できないため、次のようにポイントを保持するための IQ カタログストアテーブルを別個に作成する必要があります。

IQ メインストアテーブル	Customer( CustID, CustName, ... )
IQ カタログストアテーブル	CustPoints( CustID, Point )

ST\_Polygon P があり、P に含まれるすべての顧客を見つけるクエリが必要なシナリオを考えてみます。P は、定数値または IQ カタログストアテーブルの前のクエリの値が入力される接続変数であるとしてします。

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

このクエリは IQ メインストアの Customer テーブルを IQ カタログストアに流し込むため、パフォーマンスに影響します。CIS 機能補正のパフォーマンスに関する考慮事項が適用されます。

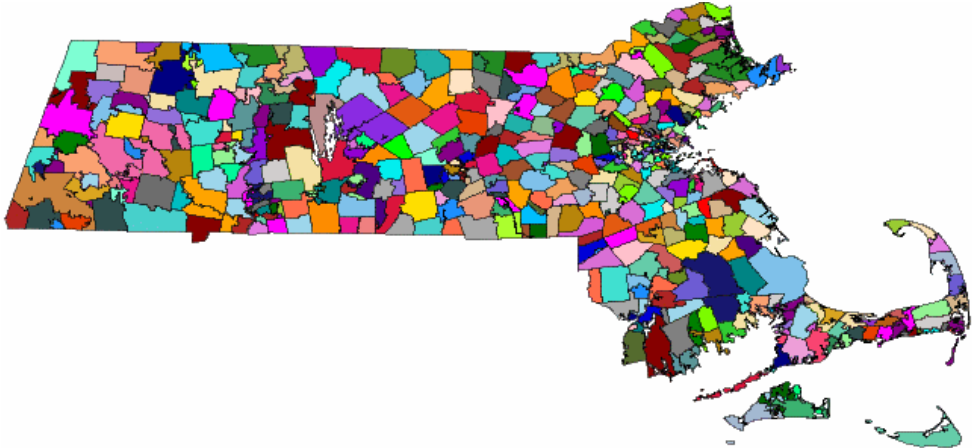
参照整合性制約は、IQ メインストア/IQ カタログストアブリッジ間で維持されません。1 つのテーブル内のロー挿入/削除は、ジョインされたテーブルに反映されます。たとえば、すべての CustPoints ローがいずれかの Customer ローに対応し

## 制限事項

ている必要があります。Customer ローを削除した場合は、対応する CustPoints ローを削除する必要があります。

# 空間データ

**空間データ**は、定義された空間内のオブジェクトの位置、シェイプ、方向を記述するデータです。SAP Sybase IQ の空間データは、ポイント、曲線(線ストリングと円弧ストリング)、多角形の形式の 2D ジオメトリとして表現されます。たとえば、次のイメージはマサチューセッツ州の郵便番号の区域を表す多角形の論理和を表しています。



空間データに対して実行される 2 つの一般的な操作は、ジオメトリ間の距離の計算と、複数のオブジェクトの論理和または共通部分の判断です。これらの計算は、Intersects、Contains、Crosses などの述部を使用して実行されます。

空間データのマニュアルでは、作業を行う空間参照系と空間データについての知識があることを前提としています。

---

**注意：** 32 ビット Windows と 32 ビット Linux の空間データサポートには、SSE2 命令をサポートする CPU が必要となります。これは、Intel Pentium 4 (2001 年リリース) 以降と AMD Opteron (2003 年リリース) 以降でサポートされます。

---

## 空間データの用例

SAP Sybase IQ の空間データサポートによって、アプリケーション開発者は既存のデータと空間情報を関連付けることができます。たとえば、会社を表すテーブルに、会社の場所をポイントとして格納したり、会社の配送地域を多角形として格納したりできます。これは、SQL では次のように表すことができます。

```
CREATE TABLE Locations(  
  ID INT,  
  ManagerName CHAR(16),  
  StoreName CHAR(16),
```

```
Address ST_Point,  
DeliveryArea ST_Polygon )
```

例で使用されている空間データ型 `ST_Point` は単一のポイントを表し、`ST_Polygon` は任意の多角形を表します。このスキーマを使用すると、アプリケーションで会社の場所をすべて地図上に表示したり、次のようなクエリを使用して会社が特定の住所に配送しているかどうかを検索したりできます。

```
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 では、空間データの格納およびデータ管理の機能が提供されているため、地理的な場所、ルート情報、シェイプデータなどの情報を格納できます。

これらの情報の断片は、ポイント、さまざまな形式の多角形、線として、対応する空間データ型 (`ST_Point`、`ST_Polygon` など) で定義されたカラムに格納されます。空間データへのアクセスおよび操作を行うには、メソッドおよびコンストラクタを使用します。SAP Sybase IQ では、他の製品との互換性のために設計された一連の SQL 空間関数も提供されます。

## 空間参照系 (SRS) と空間参照系識別子 (SRID)

空間データベースのコンテキストでは、ジオメトリが記述されている定義済みの空間を空間参照系 (SRS) と呼びます。空間参照系では、少なくとも次のことが定義されます。

- 基本となる座標系の測定単位 (角度、メートルなど)
- 座標の最大値と最小値 (境界とも呼ばれます)
- デフォルトの線形測定単位
- データが平面データまたは回転楕円体データのいずれであるか
- データを他の SRS に変換するための投影情報

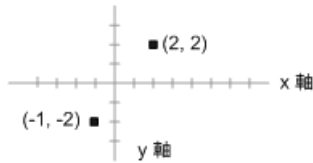
すべての空間参照系には、空間参照系識別子 (SRID) と呼ばれる識別子があります。ジオメトリが別のジオメトリと接触しているかどうかを調べる操作などを SAP Sybase IQ が実行する場合、空間参照系の計算を正しく実行できるように、SRID を使用してその空間参照系の定義を検索します。SAP Sybase IQ データベースでは、各 SRID はユニークである必要があります。

デフォルトでは、SAP Sybase IQ は次の空間参照系を新しいデータベースに追加します。

- **デフォルト - SRID 0** – これは、ジオメトリを構成するときに SQL に SRID が指定されておらず、ロードされる値にも含まれていない場合のデフォルトの空間参照系です。



デフォルトは、平らな 2 次元平面でのデータを処理する直交空間参照系です。平面上の任意のポイントは、 $x$  と  $y$  が  $-1,000,000$  から  $1,000,000$  の境界を持つ、 $x$ 、 $y$  座標の単一のペアを使用して定義されます。距離は垂直座標軸を使用して測定されます。この空間参照系には、SRID **0** が割り当てられています。



直交空間参照系は、平面タイプの空間参照系です。

- **WGS 84 - SRID 4326** – WGS 84 標準では、地球の回転楕円体の参照面が提供されます。これは、グローバルポジショニングシステム (GPS) によって使用される空間参照系です。WGS 84 の座標原点は地球の中心であり、 $\pm 1$  メートルの精度であると見なされています。WGS は世界測地系 (World Geodetic System) を表しています。

WGS 84 の座標は角度で表され、第 1 の座標は経度で境界は  $-180$  から  $180$ 、第 2 の座標は緯度で境界は  $-90$  から  $90$  です。

WGS 84 は、曲面タイプの空間参照系であり、デフォルトの測定単位は METRE です。

- **WGS 84 (平面) - SRID 1000004326** – WGS 84 (平面) は WGS 84 と似ていますが、正距円筒投影法が使用され、長さ、面積、その他の計算がゆがめられます。たとえば、SRID 4326 と 1000004326 では、どちらも赤道での経度 1 度は約  $111$  km です。北緯  $80$  度では、SRID 4326 では経度 1 度は約  $19$  km ですが、SRID 1000004326 ではすべての緯度で経度 1 度を約  $111$  km として扱います。SRID 1000004326 では長さにかかなりのゆがみが発生します ( $10$  分の  $1$  以下に減少)。ゆがみの係数は、赤道に対するジオメトリの相対的な位置によって異なります。このため、SRID 1000004326 は距離と面積の計算には使用しないでください。ST\_Contains、ST\_Touches、ST\_Covers などの関係述部のみ使用してください。

WGS 84 (平面) は、平面タイプの空間参照系であり、デフォルトの測定単位は DEGREE です。

- **sa\_planar\_unbounded - SRID 2,147,483,646** – 内部でのみ使用。
- **sa\_octahedral\_gnomonic - SRID 2,147,483,647** – 内部でのみ使用。

空間参照系は希望の方法で、および希望の任意の SRID 番号を使用して定義できるため、空間参照系の定義 (投影、座標系など) は、データがデータベース間で移動する場合、または他の SRS に変換されるときにデータに付随している必要があります。たとえば、空間データを WKT にアンロードする場合、空間参照系の定義がファイルの先頭に含まれます。

**sa\_install\_feature** システムプロシージャを使用した追加の空間参照系のインストール

SAP Sybase IQ には、使用できる定義済みの SRS も多数備えられています。ただし、これらの SRS は、新しいデータベースの作成時にデフォルトではデータベースにインストールされません。これらを追加するには、sa\_install\_feature システムプロシージャを使用してください。

これらの追加の空間参照系の説明については、[spatialreference.org](http://spatialreference.org) と [www.epsg-registry.org/](http://www.epsg-registry.org/)を参照してください。

### *現在データベースにある空間参照系のリストの決定*

空間参照系情報は、ISYSSPATIALREFERENCESYSTEM システムテーブルに格納されています。SRS の SRID は、このテーブルのプライマリキー値として使用されます。データベースサーバは、SRID 値を使用して空間参照系の設定情報を検索し、これにより、設定情報がないと抽象的な空間座標を地球上の実際の位置として解釈できるようになります。

ST\_SPATIAL\_REFERENCE\_SYSTEMS 統合ビューをクエリすることによって、空間参照系のリストを確認できます。このビューの各ローに空間参照系が定義されています。

### *一般的なマッピングアプリケーションとの互換性*

一般的な Web マッピングおよび視覚化アプリケーション (Google Earth、Bing Maps、ArcGIS Online など) には、地球の球形モデルに基づくメルカトル投影の空間参照系を使用するものがあります。この球形モデルは地球の両極での扁平を無視するため、位置で最大 800 m、縮尺で最大 0.7 パーセントの誤差が生じる可能性があります。アプリケーションでの投影実行がより効率的になります。

以前は、この空間参照系には SRID 900913 が市販アプリケーションで割り当てられていました。ただし、EPSG がこの投影を SRID 3857 としてリリースしました。SRID 900913 を要求するアプリケーションと互換性を保つために、以下のことを実行できます。

1. sa\_install\_feature システムプロシージャを使用して、SAP Sybase IQ によって提供される空間参照系をすべてインストールします (SRID 3857 を含む)。
2. dbunload -n を実行して、3857 SRID 定義を取得します (SAP Sybase IQ に dbunload ユーティリティは付属していません)。

## 測定単位

---

地理的特性は、緯度、ラジアン、またはその他の角度測定単位で測定できます。空間参照系には、地理的座標が測定される単位の名前を明示的に指定し、指定された単位からラジアンへの変換方法を含める必要があります。

投影座標系を使用している場合、個々の座標値は地球の地表面に沿った、ポイントまでの線形距離を表します。座標値は、メートル、フィート、マイル、またはヤードで測定できます。投影座標系では、座標値を表現する線形測定単位を明示的に指定する必要があります。

次の測定単位は、新しい SAP Sybase IQ データベースに自動的にインストールされます。

- **meter** – 線形測定単位。国際メートルとしても知られています。SI 標準単位です。ISO 1000 で定義されています。
- **metre** – 線形測定単位。meter のエイリアス。SI 標準単位です。ISO 1000 で定義されています。
- **radian** – 角度測定単位。SI 標準単位です。ISO 1000:1992 で定義されています。
- **degree** – 角度測定単位 ( $\pi/180.0$  ラジアン)。
- **planar degree** – 線形測定単位。60 海里として定義されています。PLANAR 線解積を使用する地理的空間参照系で使用される線形測定単位です。

### 追加の事前定義済み測定単位のインストール

sa\_install\_feature システムプロシージャは、デフォルトではインストールされていない、事前定義済みの測定単位を新しいデータベースに追加します。

#### 前提条件

なし。

#### 手順

次の文を実行して、すべての事前定義済みの測定単位をインストールします。

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

追加の測定単位はすべて、インストールされます。

#### 次のステップ

測定単位を使用する空間参照系を作成できます。

これらの追加の測定単位についての説明については、[www.epsg-registry.org/](http://www.epsg-registry.org/)を参照してください。この Web ページで、測定単位の名前を [Name] フィールドに入力

し、[Type]フィールドでUnit of Measure (UOM) を選択して、[Search]をクリックします。

## SAP Sybase IQ による空間データのサポート

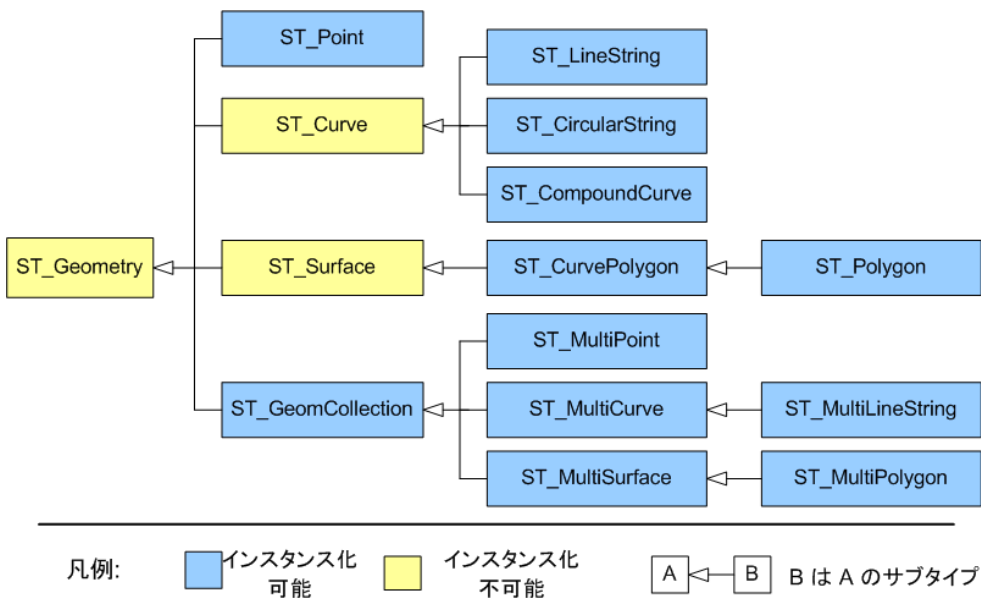
以下の項では、SAP Sybase IQ による空間データのサポートについて説明します。

### サポートされる空間データ型とその階層

SAP Sybase IQ は、地理空間データの格納とアクセスにおいて SQL Multimedia (SQL/MM) 標準に準拠しています。この標準の重要な構成要素は、ST\_Geometry 型の階層を使用して、地理空間データの作成方法を定義していることです。階層内では、プレフィクス ST がすべてのデータ型 (クラスまたはタイプとも呼ばれます) に対して使用されます。

カラムが特定のタイプとして識別されると、そのタイプとそのサブクラスの値をカラムに格納できます。たとえば、ST\_GeomCollection として識別されたカラムには、ST\_MultiPoint、ST\_MultiSurface、ST\_MultiCurve、ST\_MultiPolygon、ST\_MultiLineString の値も格納できます。

次の図は、ST\_Geometry データ型とそのサブタイプの階層を示しています。



左側のタイプがスーパータイプ (または基本タイプ)、右側がそのサブタイプ (または派生タイプ) を示します。

### サポートされる空間データ型の説明

SAP Sybase IQ では、次の空間データ型がサポートされています。

- **ポイント** – ポイントは空間内の単一の場所を定義します。ポイントジオメトリには長さまたは面積がありません。ポイントには必ず X 座標と Y 座標があります。

空でないポイントに対しては、ST\_Dimension は 0 を返します。

GIS データでは、ポイントは、通常、住所などの場所、または山などの地理的特性を表すために使用されます。

- **線ストリング** – 線ストリングは長さを持つジオメトリですが、領域はありません。空でない線ストリングに対しては、ST\_Dimension は 1 を返します。線ストリングは、単純か、単純ではないか、および閉じているか、閉じていないかによって特徴付けることができます。**単純**とは、それ自体が交差していない線ストリングを指します。**閉じている**とは、開始したポイントと同じポイントで終了する線ストリングを意味します。たとえば、リングは単純で閉じている線ストリングの例です。

GIS データでは、線ストリングは、通常、河川、道路、または配送経路を表すために使用されます。

- **多角形** – 多角形は空間の領域を定義します。多角形は、外部領域を定義する 1 つの外部境界リングと、領域の穴を定義する 0 個以上の内部リングによって構成されます。多角形には領域が関連付けられますが、長さはありません。

空でない多角形に対しては、ST\_Dimension は 2 を返します。

GIS データでは、多角形は、通常、地域(国、町、州など)、湖、公園などの大きな地理的特性を表すために使用されます。

- **円ストリング** – 円ストリングは、一連の円弧セグメントを接続したものです。ポイント間が円弧で接続された線セグメントとよく似ています。
- **複合曲線** – 複合曲線は、円ストリングまたは線ストリングを接続したものです。
- **曲線多角形** – 曲線多角形は、一般的な多角形であり、円弧の境界セグメントが含まれる場合もあります。
- **ジオメトリ** – ジオメトリという語は、ポイント、線ストリング、多角形などのオブジェクトを包含するタイプを意味します。ジオメトリタイプは、サポートされるすべての空間データ型のスーパータイプです。
- **ジオメトリコレクション** – ジオメトリコレクションは、1 つ以上のジオメトリ(ポイント、線、多角形など)のコレクションです。
- **複数ポイント** – 複数ポイントは個々のポイントのコレクションです。

GIS データでは、複数ポイントは、通常、ロケーションのセットを表すために使用されます。

- **複数多角形** – 複数多角形は 0 個以上の多角形の集合体です。

GIS データでは、複数多角形は、複数の島で形成されている国の領土や、湖水系などの地理的特性を表現するためによく使用されます。

- **複数線ストリング** – 複数線ストリングは線ストリングの集合体です。

GIS データでは、複数線ストリングは、河川または高速道路網のような地理的特性を表すために使用されます。

- **複数面** – 複数面は曲線多角形の集合体です。

### 空間データ型のオブジェクト指向型プロパティ

- サブタイプ (または派生タイプ) は、スーパータイプ (または基本タイプ) よりも限定的です。たとえば、ST\_LineString は ST\_Curve をより限定的にしたものです。
- サブタイプはすべてのスーパータイプのすべてのメソッドを継承します。たとえば、ST\_Polygon の値は、スーパータイプ ST\_Geometry、ST\_Surface、ST\_CurvePolygon のメソッドを呼び出せます。
- サブタイプの値は自動的に派生元のスーパータイプの値に変換されます。たとえば、point1.ST\_Distance( point2 ) のように、ST\_Geometry のパラメータが必要な場面では、ST\_Point の値を使用できます。
- カラムまたは変数には、どのサブタイプの値でも格納できます。たとえば、ST\_Geometry (SRID=4326) タイプのカラムには、すべてのタイプの空間値を格納できます。
- 宣言されたタイプを使用したカラム、変数、または式は、そのままの型で処理できます。そうでない場合は、サブタイプにキャストします。たとえば、geom という名前の ST\_Geometry のカラムの ST\_Polygon の値を、TREAT 関数で宣言した ST\_Surface タイプに変更することで、TREAT( geom AS ST\_Surface ).ST\_Area() のように、ST\_Area メソッドを呼び出すことができます。

### サポートされる空間述部

述部は条件式であり、論理演算子 AND や OR と組み合わせて、WHERE 句、HAVING 句、ON 句、IF 式、CASE 式、または CHECK 制約に一連の条件を構成します。SQL では、述部は TRUE または FALSE に評価できます。多くのコンテキストで、UNKNOWN と評価される述部が FALSE として解釈されます。

空間述部は、0 または 1 を返すメンバー関数として実装されます。空間述部をテストするには、クエリで = または <> 演算子を使用して、関数の結果を 1 または 0 で比較します。次に例を示します。

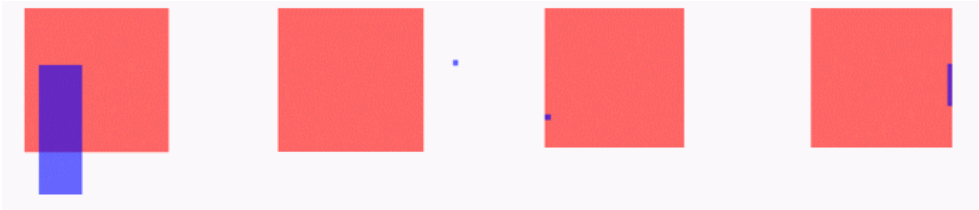
```
SELECT * FROM SpatialShapes WHERE geometry.ST_IsEmpty() = 0;
```

述部を使用するのは、次のような質問で空間データを問い合わせる場合です。たとえば、2つ以上のジオメトリの距離がどのくらい近いか、それらは交差しているのか、重なり合っているか、あるいは、あるジオメトリが別のジオメトリ内に含まれているか、などがあります。たとえば、配送会社の場合、述部を使用して、顧客が特定の配送区域内に存在しているかどうかを判別できます。

### 空間述部の直感性

述部の結果は直感的ではないことがあるため、特殊なケースをテストして、望みどおりの結果を得られていることを確認する必要があります。たとえば、ジオメトリが別のジオメトリを包含するには (a.ST\_Contains(b)=1)、またはジオメトリが別のジオメトリ内に含まれるためには (b.ST\_Within(a)=1)、a の内部と b の内部は交差している必要があり、b のどの部分も a の外部と交差することはできません。ただし、ジオメトリが別のジオメトリ内に含まれていることを予想していたが、含まれていない場合があります。

たとえば、次の例では、a.ST\_Contains(b) と b.ST\_Within(a) (a が赤) に対して 0 が返されます。



ケース 1 と 2 は明白です。紫色のジオメトリは赤い正方形内に完全には含まれていません。ケース 3 と 4 は明らかではありません。これらのケースは、紫色のジオメトリが赤い正方形の境界上のみあります。紫色のジオメトリは赤い正方形内にあるように見えますが、ST\_Contains は赤い正方形内にあるとは見なしません。

ST\_Covers と ST\_CoveredBy は、ST\_Contains と ST\_Within と類似した述部です。ST\_Covers と ST\_CoveredBy の場合、2つのジオメトリの内部が交差している必要がない点が異なります。また、ST\_Covers と ST\_CoveredBy が返す結果は、ST\_Contains と ST\_Within より直感的でわかりやすい場合が多いです。

述部のテストで望んでいたものと異なる結果が返された場合、ST\_Relate メソッドを使用して、テストする関係を厳密に指定することを検討してください。

## 空間標準への準拠

SAP Sybase IQ の空間は、次の標準に準拠しています。

- **国際標準化機構 (ISO)** – SAP Sybase IQ のジオメトリは、空間のユーザデータ型、ルーチン、スキーマの定義と空間データの処理において、ISO 標準に準拠して

います。SAP Sybase IQ は、国際標準 ISO/IEC 13249-3:2006 に記述されている特定の推奨事項に準拠しています。詳細については、[http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=38651](http://www.iso.org/iso/catalogue_detail.htm?csnumber=38651)を参照してください。

- **OGC (Open Geospatial Consortium) ジオメトリモデル** – SAP Sybase IQ のジオメトリは、「OGC OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option version 1.2.0 (OGC 06-104r3)」に準拠しています。<http://www.opengeospatial.org/standards/sfs> を参照してください。

SAP Sybase IQ では、OGC 推奨の標準を使用して、空間情報をさまざまなベンダーおよびアプリケーション間で共有できるようにしています。

SAP Sybase IQ の空間ジオメトリと互換性を確保するには、OGC によって指定されている標準に準拠することをおすすめします。

- **SQL Multimedia (SQL/MM)** – SAP Sybase IQ は SQL/MM 標準に準拠し、すべてのメソッド名と関数名にプレフィクス `ST_` を使用しています。

SQL/MM は、SQL を使用して空間データを格納、検索、処理する方法を定義した国際標準です。空間データ型の階層 (`ST_Geometry` など) は、空間データの検索に使用される方法の 1 つです。`ST_Geometry` 階層には、`ST_Point`、`ST_Curve`、`ST_Polygon` などの多くのサブタイプが含まれています。SQL/MM 標準では、クエリに含まれるすべての空間値は、同じ空間参照系に定義されている必要があります。

## サポートと準拠についての特記事項

この項では、SAP Sybase IQ の空間データサポートの特記事項 (サポートされない機能、他のデータベース製品との顕著な動作の違いなど) について説明します。

- **ジオグラフィとジオメトリ** – ベンダーによっては、**ジオグラフィ** (曲面上のオブジェクトに関連する) か、**ジオメトリ** (平面上のオブジェクト) かによって、空間オブジェクトが区別されます。SAP Sybase IQ では、空間オブジェクトはすべてジオメトリと見なされ、オブジェクトの SRID によって、曲面空間参照系または平面空間参照系で処理されていることが示されます。
- **サポートされないメソッド** –

`ST_Buffer` メソッド

`ST_LocateAlong` メソッド

`ST_LocateBetween` メソッド

`ST_Segmentize` メソッド

`ST_Simplify` メソッド

`ST_Distance_Spheroid` メソッド

`ST_Length_Spheroid` メソッド



## サポートされる空間データのインポートフォーマットとエクスポートフォーマット

次の表に、SAP Sybase IQ によってサポートされる、空間データのインポートおよびエクスポートのためのデータフォーマットとファイルフォーマットをリストします。

データフォーマット	インポート	エクスポート	説明
WKT (Well Known Text)			<p>ASCII テキストで表現された地理的データ。このフォーマットは、地理的情報の OpenGIS 実装仕様に定義されている単純特性の一部として OGC (Open Geospatial Consortium) によって保守されています。www.opengeospatial.org/standards/sfa を参照してください。</p> <p>WKT でポイントを表す方法の例を次に示します。</p> <pre>'POINT(1 1)'</pre>
WKB (Well Known Binary)			<p>バイナリストリームとして表現された地理的データ。このフォーマットは、地理的情報の OpenGIS 実装仕様に定義されている単純特性の一部として OGC によって保守されています。www.opengeospatial.org/standards/sfa を参照してください。</p> <p>WKB でポイントを表す方法の例を次に示します。</p> <pre>'010100000000000000000000F03F000000000000F03F'</pre>
EWKT (拡張 Well Known Text)			<p>SRID 情報が埋め込まれた WKT フォーマット。このフォーマットは、PostGIS (PostgreSQL の空間データベース拡張) の一部として管理されます。postgis.refrations.net/ を参照してください。</p> <p>EWKT でポイントを表す方法の例を次に示します。</p> <pre>'srid=101;POINT(1 1)'</pre>
EWKB (拡張 Well Known Binary)			<p>SRID 情報が埋め込まれた WKB フォーマット。このフォーマットは、PostGIS (PostgreSQL の空間データベース拡張) の一部として管理されます。postgis.refrations.net/ を参照してください。</p> <p>EWKB でポイントを表す方法の例を次に示します。</p> <pre>'01010000020040000000000000000000F03F000000000000F03F'</pre>

データフォーマット	インポート	エクスポート	説明
GML (Geographic Markup Language)	×		<p>地理的空間データを表すために使用される XML 文法。この標準は OGC (Open Geospatial Consortium) によって保守されており、インターネットを介した地理的データの交換を目的としています。 <a href="http://www.opengeospatial.org/standards/gml">www.opengeospatial.org/standards/gml</a> を参照してください。</p> <p>GML でポイントを表す方法の例を次に示します。</p> <pre>&lt;gml:Point&gt; &lt;gml:coordinates&gt;1,1&lt;/gml:coordinates&gt; &lt;/gml:Point&gt;</pre>
KML	×		<p>以前の Google の Keyhole Markup Language です。この XML 文法は地理的データを表すために使用され、視覚化、ナビゲーション補助、地図とイメージに注釈を付ける機能を含んでいます。Google はこの標準を OGC に提案しました。OGC はこれをオープン標準として受け入れ、現在は KML と呼ばれています。 <a href="http://www.opengeospatial.org/standards/kml">www.opengeospatial.org/standards/kml</a> を参照してください。</p> <p>KML でポイントを表す方法の例を次に示します。</p> <pre>&lt;Point&gt; &lt;coordinates&gt;1,0&lt;/coordinates&gt; &lt;/Point&gt;</pre>
ESRI シェイプファイル		×	<p>シェイプファイル (シェイプを定義するために一緒に使用される複数のファイル) の形式で空間オブジェクトを表すための一般的な地理空間ベクトルデータフォーマット。</p>
GeoJSON	×		<p>名前/値ペア、値の順序付きリスト、一般的なプログラミング言語 (C、C++、C#、Java、JavaScript、Perl、Python など) で使用される規則に類似した規則を使用するテキストフォーマット。</p> <p>GeoJSON は JSON 標準のサブセットであり、地理的情報をコード化するために使用されます。SAP Sybase IQ では、GeoJSON 標準をサポートしており、SQL 出力を GeoJSON フォーマットに変換するための ST_AsGeoJSON メソッドを提供しています。</p> <p>GeoJSON でポイントを表す方法の例を次に示します。</p> <pre>{"x" : 1, "y" : 1, "spatialReference" : {"wkid" : 4326}}</pre> <p>GeoJSON 仕様の詳細については、 <a href="http://geojson.org/geojson-spec.html">geojson.org/geojson-spec.html</a> を参照してください。</p>

データフォーマット	インポート	エクスポート	説明
SVG (Scalable Vector Graphic) ファイル	×		<p>2次元のジオメトリを表すために使用される XML ベースのフォーマット。SVG フォーマットは W3C (World Wide Web Consortium) によって保守されています。www.w3.org/Graphics/SVG/を参照してください。</p> <p>SVG でポイントを表す方法の例を次に示します。</p> <pre>&lt;rect width="1" height="1" fill="deepskyblue" stroke="black" stroke-width="1" x="1" y="-1"/&gt;</pre>

## ESRI シェイプファイルのサポート

SAP Sybase IQ では、ESRI (Environmental System Research Institute) のシェイプファイルフォーマットをサポートします。ESRI シェイプファイルは、データセット内の空間機能のジオメトリデータと属性情報を格納するために使用されます。

ESRI シェイプファイルには、少なくとも異なる種類の 3 つのファイル (.shp、.shx、および .dbf) が含まれています。メインファイルの拡張子は .shp、インデックスファイルの拡張子は .shx、属性カラムの拡張子は .dbf です。すべてのファイルは同じベース名を共有し、多くの場合単一の圧縮ファイルにまとめられます。SAP Sybase IQ では、MultiPatch を除くすべてのシェイプタイプの ESRI シェイプファイルを読み込むことができます。これには、Z データと M データを含んだシェイプタイプが含まれます。

ESRI シェイプファイル内のデータには、通常、複数のローとカラムが含まれています。たとえば、空間チュートリアルでは、マサチューセッツ州の郵便番号区域が含まれるシェイプファイルをロードします。このシェイプファイルには、郵便番号区域ごとに 1 つのローがあり、ローにはその区域の多角形情報が含まれています。また、各郵便番号区域の追加の属性 (カラム) も含んでいます。これには、郵便番号名 (たとえば、文字列「02633」) やその他の属性が含まれています。

テーブルにシェイプファイルをロードする最も簡単な方法は、Interactive SQL の [インポートウィザード] または st\_geometry\_load\_shapefile システムプロシージャを使用することです。どちらのツールを使用しても、適切なカラムのテーブルが作成され、シェイプファイルのデータがロードされます。

また、LOAD TABLE 文と INPUT 文を使用することによってもシェイプファイルをロードできますが、その場合、ロード操作を実行する前に、適切なカラムのテーブルが作成されている必要があります。

## 空間データ

LOAD TABLE 文または INPUT 文を使用してデータをロードするときに必要なカラムを知るために、sa\_describe\_shapefile システムプロシージャを使用できます。

ESRI シェイプファイルの詳細については、<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> を参照してください。

## 空間トピック関連の推奨ドキュメント

地表面のマッピングおよび測定 (測地学) に使用される別の方法についての入門書、座標 (または空間) 参照系に関連する主要な概念については、[www.epsg.org/guides/index.html](http://www.epsg.org/guides/index.html) にアクセスして、[Geodetic Awareness] を選択してください。

地理的情報に関する OGC OpenGIS 実装仕様 - 単純地物アクセス:

[www.opengeospatial.org/standards/sfs](http://www.opengeospatial.org/standards/sfs)

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

SVG (Scalable Vector Graphics) 1.1 仕様: [www.w3.org/Graphics/SVG/](http://www.w3.org/Graphics/SVG/)

GML (Geographic Markup Language) 仕様: [www.opengeospatial.org/standards/gml](http://www.opengeospatial.org/standards/gml)

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

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

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

## 空間カラムの作成 (SQL の場合)

空間データをサポートするカラムを追加して、空間データを任意のテーブルに追加できます。

### 前提条件

そのテーブルの所有者であるか、そのテーブルに対する ALTER 権限を持っているか、または ALTER ANY TABLE または ALTER ANY OBJECT システム権限を持っていることが必要です。

### 手順

1. データベースに接続します。
2. ALTER TABLE 文を実行します。

空間カラムが、既存のテーブルに追加されます。

### 次のステップ

カラムに SRID 制約を課し、空間カラムに保存できる値を制限できます。

## 空間カラムのインデックス

---

空間インデックスを作成するときは、他のデータ型のインデックスの作成と同様に、CREATE INDEX 文または [テキストインデックス作成ウィザード] を使用します。ただし、空間データに対してインデックスを作成する場合は、複数の空間カラムをインデックスに含めないことと、空間カラムをインデックス定義の最後に配置することをおすすめします。

空間カラムをインデックスに含めるには、カラムに SRID 制約を付ける必要もあります。

空間データにインデックスを作成すると、ジオメトリ間の関係を評価する場合のコストを低減できます。たとえば、販売区域の境界を変更することを検討しており、そのことが既存の顧客に与える影響を判別するとします。提案された販売区域内に存在する顧客を判別するには、ST\_Within メソッドを使用して、各顧客の住所を表すポイントと販売区域を表す多角形を比較します。インデックスがない場合、データベースサーバは、Customer テーブルのすべての住所のポイントを販売区域の多角形に対してテストし、結果で返すかどうか判別します。この操作は、Customer テーブルが大きい場合は高コストになり、販売区域が小さい場合は非効率になります。各顧客の住所のポイントが含まれるインデックスを使用すると、より短時間で結果を返すことができます。販売区域とそれに重なり合う州を関連付ける述部をクエリに追加できる場合、州コードと住所ポイントの両方が含まれるインデックスを使用して、結果をより短時間で取得できることがあります。

空間クエリは、クラスタドインデックスによって効率がよくなる場合がありますが、テーブルのその他の用途を考慮してから、クラスタドインデックスの使用を決定する必要があります。実行される可能性のあるクエリのタイプを検討およびテストして、クラスタドインデックスによってパフォーマンスが向上するかどうかを判別します。

空間カラムに対してテキストインデックスを作成できますが、通常のインデックスと比較して利点はありません。代わりに、通常のインデックスの使用をおすすめします。

---

**注意：**空間カラムには、プライマリーキー、ユニークインデックス、一意性制約のいずれも入れることはできません。

---

## 空間データ型の構文

---

SQL/MM 標準では、ANSI/SQL CREATE TYPE 文に基づいて構築される拡張ユーザ定義型 (UDT) として空間データのサポートが定義されています。SAP Sybase IQ で

はユーザ定義型はサポートされませんが、サポートされているかのように SAP Sybase IQ の空間データサポートが実装されています。

### UDT インスタンスのインスタンス化

次のように、コンストラクタを呼び出すことによって、ユーザ定義型の値をインスタンス化できます。

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

たとえば、クエリに次のように指定して、2つの ST\_Point 値をインスタンス化できます。

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

SAP Sybase IQ では、通常のオーバーロード解決ルールを使用して、*argument-list* を定義済みコンストラクタと照合します。次の状況では、エラーが返されます。

- NEW がユーザ定義型ではないタイプで使用された場合
- ユーザ定義型がインスタンス化可能ではない場合 (たとえば、ST\_Geometry はインスタンス化可能なタイプではありません)
- 指定された引数型と一致するオーバーロードがない場合

### インスタンスメソッドの使用

ユーザ定義型には、インスタンスメソッドが定義されていることがあります。インスタンスメソッドは、次のようにしてタイプの値に対して呼び出されます。

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

たとえば、次の架空の例では、Massdata.CenterPoint カラムの X 座標を SELECT しています。

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

CenterPoint というユーザ ID がある場合、データベースサーバでは CenterPoint.ST\_X() があいまいであると見なされます。これは、この文が「ユーザ CenterPoint が所有するユーザ定義関数 ST\_X を呼び出す」(この文の意図しない意味)とも、「Massdata.CenterPoint カラムの ST\_X メソッドを呼び出す」(正しい意味)とも考えられるためです。データベースサーバは、最初に CenterPoint という名前のユーザに対して、大文字小文字を区別した検索を実行することによって、このようなあいまいさを解決します。ユーザが見つかった場合、データベースサーバは、ユーザ CenterPoint が所有する ST\_X というユーザ定義関数を呼び出していると思なして処理を続行します。見つからなかった場合には、データベースサーバは、構成体をメソッド呼び出しとして処理し、Massdata.CenterPoint カラムの ST\_X メソッドを呼び出します。

次の場合、インスタンスメソッド呼び出しはエラーを返します。

- *value-expression* の宣言されたタイプがユーザ定義型ではない

- *value-expression* またはそのスーパータイプの 1 つの宣言されたタイプに、名前付きメソッドが定義されていない
- *argument-list* が、名前付きメソッドに対して定義されたオーバーロードのいずれとも一致しない

#### 静的メソッドの使用

ANSI/SQL 標準では、インスタンスメソッドに加えて、ユーザ定義型に静的メソッドを関連付けることができます。静的メソッドは次の構文を使用して呼び出されます。

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

たとえば、次の文はテキストを解析することによって ST\_Point をインスタンス化します。

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

次の場合、静的メソッド呼び出しはエラーを返します。

- *value-expression* の宣言されたタイプがユーザ定義型ではない
- *value-expression* またはそのスーパータイプの 1 つの宣言されたタイプに、名前付きメソッドが定義されていない
- *argument-list* が、名前付きメソッドに対して定義されたオーバーロードのいずれとも一致しない

#### 静的集約メソッドの使用 (SAP Sybase IQ 拡張)

ANSI/SQL の拡張として、SAP Sybase IQ では、ユーザ定義の集約を実装する静的メソッドをサポートしています。次に例を示します。

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

静的メソッドのオーバーロードは、すべてが集約であるか、またはすべてが集約ではないかのいずれかである必要があります。

次の場合、静的集約メソッド呼び出しはエラーを返します。

- 静的メソッド呼び出しがエラーを返す場合
- 組み込み集合関数がエラーを返す場合
- WINDOW 句が指定されている場合

#### 型述部の使用

ANSI/SQL 標準では、文で値の具象タイプ (他の言語ではオブジェクトタイプとも呼ばれる) を検査できる型述部が定義されています。構文は次のとおりです。

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

*value* が NULL の場合、述部は UNKNOWN を返します。それ以外の場合、*value* の具象タイプが *type-name* リストの各要素と比較されます。ONLY が指定されている場合、具象タイプが指定されたタイプと同一である場合に一致と見なされます。

それ以外の場合は、具象タイプが指定されたタイプまたは派生タイプ (サブタイプ) である場合に一致と見なされます。

*value* の具象タイプがリスト内のいずれかの要素と一致した場合は TRUE が返され、それ以外の場合は FALSE が返されます。

次の例では、Shape カラムの値に具象タイプ ST\_Curve またはそのサブタイプの 1 つ (ST\_LineString、ST\_CircularString、または ST\_CompoundCurve) が含まれているすべてのローを返します。

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

### サブタイプへの TREAT 式の使用

ANSI/SQL 標準では、式の宣言されたタイプをスーパータイプからサブタイプに効率的に変換できる、サブタイプ処理式が定義されています。式の具象タイプ (別の言語ではオブジェクトタイプとも呼ばれる) が指定したサブタイプ、または指定したサブタイプのサブタイプであることがわかっている場合には、この式を使用できます。CAST 関数で値のコピーが作成されるのに対して、TREAT ではコピーが作成されないため、これは CAST 関数を使用するよりも効率的な方法です。構文は次のとおりです。

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

エラー状況が発生しない場合、結果は *target-subtype* で宣言されたタイプの *value-expression* となります。

次の場合、サブタイプ処理式はエラーを返します。

- *value-expression* がユーザ定義型ではない場合
- *target-subtype* が *value-expression* の宣言されたタイプのサブタイプではない場合
- *value-expression* の動的タイプが *target-subtype* のサブタイプではない場合

次の例では、ST\_Geometry の Shape カラムの宣言されたタイプを効率的な ST\_Curve サブタイプに変更することによって、ST\_Curve タイプの ST\_Length メソッドを呼び出せるようにしています。

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

## ジオメトリの作成方法

データベースにジオメトリを作成するには、いくつかの方法があります。

- **WKT (Well Known Text) フォーマット**または **WKB (Well Known Binary) フォーマット**からのロード - WKT フォーマットまたは WKB フォーマットのデータをロードまたは挿入できます。これらのフォーマットは OGC によって定義されており、すべての空間データベースベンダーがサポートしています。SAP



Sybase IQ では、これらのフォーマットからジオメトリタイプへの自動的な変換が実行されます。

- **ESRI シェイプファイルからのロード** – ESRI シェイプファイルのデータを新規または既存のテーブルにロードできます。これを実行する方法は数多くあります。
- **SELECT...FROM OPENSTRING 文の使用** – 空間データを含むファイルで SELECT... FROM OPENSTRING 文を実行できます。次に例を示します。

```
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 )
```

- **緯度値と経度値を組み合わせた座標ポイントの作成** – 緯度と経度のデータを組み合わせて、空間データ型 ST\_Point の座標を作成できます。たとえば、緯度と経度のカラムをすでに持つテーブルがある場合、次のような文を使用して、ポイントとして値を保持する ST\_Point カラムを作成できます。

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

- **コンストラクタと静的メソッドを使用したジオメトリの作成** – コンストラクタと静的メソッドを使用してジオメトリを作成できます。

## 空間データのイメージとしての表示 (Interactive SQL の場合)

Interactive SQL では、[空間プレビュー] タブを使用してジオメトリをイメージとして表示し、データベースのデータが表すものを理解できます。

### 前提条件

選択元となるテーブルに対する SELECT 権限、または SELECT ANY TABLE システム権限が必要です。

### 手順

Interactive SQL の各インスタンスは、データベースへのそれぞれの接続に関連付けられています。Interactive SQL 内から [空間ビューア] のインスタンスを開くと、その [空間ビューア] のインスタンスは Interactive SQL のインスタンスと関連付けられたままであり、データベースへの接続を共有します。

[空間ビューア] でクエリを実行しているときに、関連付けられている Interactive SQL のインスタンスでクエリを実行しようとする、エラーが発生します。同様

に、Interactive SQL の同じインスタンスによって作成された複数の [空間ビューア] のインスタンスを開いている場合、クエリを実行できるのは、それらのインスタンスのいずれか 1 つのみです。その他のインスタンスは、そのクエリが完了するのを待機する必要があります。

---

**注意：** デフォルトでは、Interactive SQL は [結果] ウィンドウ枠内の値を 256 文字にトランケートします。Interactive SQL が完全なカラム値を読み込めないことを示すエラーを返す場合、トランケーション値を増やします。これを行うには、[ツール] » [オプション] をクリックし、左ウィンドウ枠で [SAP Sybase IQ] をクリックします。[結果] タブで、[トランケーションの長さ] を 5000 などの大きい値に変更します。[OK] をクリックして変更を保存し、再びクエリを実行してから、再度そのローをダブルクリックします。

---

1. Interactive SQL のデータベースに接続します。
2. クエリを実行して、テーブルから空間データを選択します。次に例を示します。

```
SELECT * FROM owner.spatial-table;
```

3. [結果] ウィンドウ枠の Shapes カラムの任意の値をダブルクリックして、値を [値] ウィンドウに表示します。

値は、[値] ウィンドウの [テキスト] タブにテキストとして表示されます。

4. [空間プレビュー] タブをクリックし、ジオメトリを SVG (Scalable Vector Graphic) として表示します。

ジオメトリは、Scalable Vector Graphic (SVG) として表示されます。

### 次のステップ

[前のロー] と [次のロー] ボタンを使用して結果セットの他のローを表示することで、空間データをジオメトリとして表示できます。

## 空間データのイメージとしての表示 (Spatial Viewer の場合)

---

空間ビューアを使用して複数のジオメトリをイメージとして表示し、データベースのデータが表すものを理解できます。

### 前提条件

選択元となるテーブルに対する SELECT 権限、または SELECT ANY TABLE システム権限が必要です。

## 手順

画像はローが処理される順序で描画され、最新の画像が最上位に表示されるため、結果のローの順序は画像が[空間ビューア]に表示される方法に影響を与えます。結果セットでは、後で出現するシェイプが先に出現したシェイプを覆い隠します。

1. Interactive SQL のデータベースに接続し、[ツール] » [空間ビューア] をクリックします。
2. [空間ビューア] の [SQL] ウィンドウ枠で、次のようなクエリを実行し、[実行] をクリックします。

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

3. [塗りつぶしなしでポリゴンの描画] ツールを使用すると、図形のポリゴンから色彩を削除して、すべてのシェイプのアウトラインを表示します。このツールは、保存、ズーム、パンのコントロールの近くにあるイメージの下にあります。

結果セットのすべてのジオメトリは、1つのイメージとして [結果] 領域に表示されます。

## 次のステップ

なし。

## WKT (Well Known Text) ファイルからの空間データのロード

空間データをデータベースにロードし、ジオメトリとして表示できるテキストを含んだ Well Known Text (WKT) ファイルを使用して、空間データをテーブルに追加できます。

### 前提条件

データをロードするために必要な権限は、-gl サーバオプションによって異なります。-gl オプションが ALL に設定されている場合、次のいずれかの条件に該当する必要があります。

そのテーブルの所有者である。

そのテーブルに対する LOAD 権限を持っている。

LOAD ANY TABLE システム権限を持っている。

ALTER ANY TABLE システム権限を持っている。

-gl オプションが DBA に設定されている場合は、LOAD ANY TABLE または ALTER ANY TABLE システム権限が必要です。

-gl オプションが NONE に設定されている場合、LOAD TABLE は使用できません。

## 空間データ

クライアントコンピュータのファイルからロードする場合、次の条件が必要になります。

- READ CLIENT FILE 権限も必要。
- 読み込み元のディレクトリに対する読み込み権限が必要。
- allow\_read\_client\_file データベースオプションが有効であることが必要。
- read\_client\_file セキュリティ機能が有効であることが必要。

### 手順

1. データベースにロードできる空間データを WKT フォーマットに含めたファイルを作成します。

ファイルは、LOAD TABLE 文によってサポートされるフォーマットです。

2. Interactive SQL で、データベースに接続します。
3. 次のような文を使用して、テーブルを作成し、ファイルからデータをロードします。

```
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 ',';
```

データがテーブルにロードされます。

空間データは、WKT ファイルから正常にロードされます。

### 次のステップ

[空間ビューア]を使用すると、Interactive SQL のデータを表示できます。

## 空間参照系の作成または管理

SAP Sybase IQ の空間参照単位を作成および管理するには、Interactive SQL または SAP Control Center を使用します。

SRS に関連付ける測定単位は、あらかじめ用意しておく必要があります。

SAP Control Center では、空間参照系を作成するときに既存の空間参照系をテンプレートとして使用し、その設定を編集することができます。このため、作成したい空間参照系と類似した空間参照系を選択してください。

空間参照系を作成するには、次のいずれかが必要です。

- `MANAGE ANY SPATIAL OBJECT` システム権限
- `CREATE ANY OBJECT` システム権限

空間参照系を変更するには、次のいずれかが必要です。

- その空間参照系の所有者である
- 空間参照系に対する `ALTER` 権限
- `MANAGE ANY SPATIAL OBJECT` システム権限
- `ALTER ANY OBJECT` システム権限

空間参照系を削除するには、次のいずれかが必要です。

- `MANAGE ANY SPATIAL OBJECT` システム権限
- `DROP ANY OBJECT` システム権限
- 空間参照系を所有している。

## 空間測定単位の作成または管理

---

複数の測定単位がインストールされています。インストールされた測定単位がデータに適切でない場合、独自の測定単位を作成できます。

SAP Sybase IQ の空間測定単位を作成および管理するには、`Interactive SQL` または `SAP Control Center` を使用します。

空間測定単位を作成するには、次のいずれかが必要です。

- `MANAGE ANY SPATIAL OBJECT` システム権限
- `CREATE ANY OBJECT` システム権限

空間測定単位を削除するには、次のいずれかが必要です。

- `MANAGE ANY SPATIAL OBJECT` システム権限
- `DROP ANY OBJECT` システム権限
- 空間測定単位を所有している。



# 空間に関する高度なトピック

この項では、空間に関する高度なトピックについて説明します。

## 平面および曲面の表現方法

---

SAP Sybase IQ では、平面と曲面の両方の表現がサポートされます。**平面参照系**では、地表全体またはその一部が平坦な 2 次元平面に投影され、単純な 2D ユークリッド幾何学が使用されます。ポイント間の線は直線 (円ストリングを除く) であり、ジオメトリは平面の端をまたぐ (日付変更線を越える) ことはできません。

**曲面空間参照系**は、楕円を使用して地球を表します。ポイントは計算のために楕円にマッピングされ、すべての線は最短の経路をたどり、円弧は極に向かっていきます。ジオメトリは日付変更線をまたぐことができます。

平面と曲面の表現にはそれぞれ制限があります。地球のすべての特性を最適に表す完璧な地図投影法は 1 つもありません。オブジェクトの地球上での位置によって、ゆがみが面積、シェイプ、距離、または方向に影響することがあります。

### *曲面空間参照系の制限*

曲面空間参照系 (WGS 84 など) を使用する場合、利用できない操作が多数あります。たとえば、距離の計算は、ポイントまたはポイントのコレクションに制限されます。

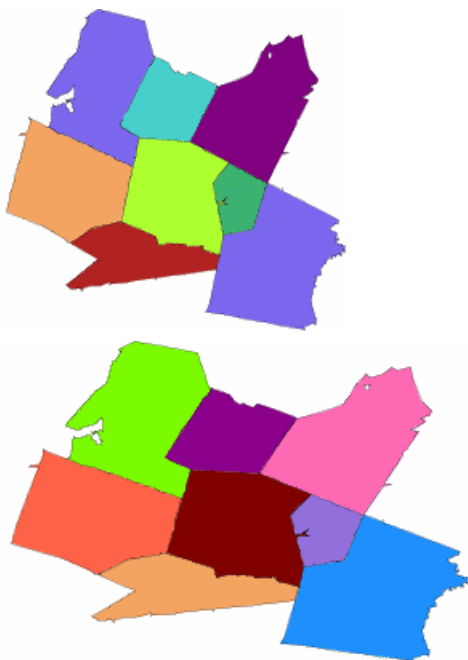
いくつかの述部と集合操作も利用できません。

円ストリングは、曲面の空間参照系では使用できません。

曲面空間参照系での計算は、対応する平面空間参照系での計算より高コストになります。

### *平面空間参照系の制限*

平面空間参照系は、定義された投影を持つ平面の空間参照系です。**投影**を使用すると、平面空間参照系を使用して曲面データを操作する場合に発生するゆがみの問題が解決されます。投影が使用されない場合に発生するゆがみの例として、次の 2 つのイメージはマサチューセッツ州の同じ郵便番号区域のグループを示しています。最初のイメージは、データを SRID 3586 で表現した投影平面空間参照系であり、マサチューセッツ州のデータを示しています。2 番目のイメージは、投影なしに平面空間参照系 (SRID 1000004326) でデータを表現しています。ゆがみは 2 番目のイメージに現れています。距離、長さ、および面積が実際より大きく、イメージが水平方向に引き伸ばされたように見えます。



他にも平面空間参照系で可能な計算はありますが、投影が影響するため、正確に計算できるのは境界に区切られたサイズの面積のみです。

作業対象が数百キロメートル以内の距離であれば、曲面データを平面空間参照系に投影して、適度な精度で距離の計算を実行できます。平面投影空間参照系にデータを投影するには、ST\_Transform メソッドを使用します。

## 「グリッドにスナップ」と許容度が空間の計算に与える影響

グリッドにスナップは、グリッド上の交差点に合うように、ジオメトリのポイントを位置付けるアクションです。グリッドに位置付けるときは、四捨五入と同じように、X と Y の値がわずかに移動される場合があります。空間データのコンテキストでは、グリッドは、空間参照系の 2 次元表現上に定義された線のフレームワークです。SAP Sybase IQ では正方形のグリッドを使用します。

グリッドにスナップの最も簡単な例として、たとえば、グリッドサイズが 0.2 の場合、Point( 14.2321, 28.3262 ) と Point( 15.3721, 27.1128 ) を結ぶ直線は、Point( 14.2, 28.4 ) と Point( 15.4, 27.2 ) を結ぶ直線にスナップされます。通常、グリッドサイズはこの単純な例よりも小さいため、精度が失われる可能性はずっと低くなります。

デフォルトでは、SAP Sybase IQ は空間参照系の X 境界と Y 境界内のすべてのポイントに対して 12 有効桁数を格納できるようにグリッドサイズを自動的に設定しま



す。たとえば、X 値の範囲が -180 から 180 まで、Y 値の範囲が -90 から 90 までの場合、データベースサーバはグリッドサイズを  $1e-9$  (0.000000001) に設定します。つまり、水平と垂直のグリッド線間の距離が  $1e-9$  となります。グリッド線の交差ポイントは、空間参照系で表現できるすべてのポイントを表します。ジオメトリが作成またはロードされると、各ポイントの X 座標と Y 座標が、グリッド上の最も近いポイントにスナップされます。

**許容度**は、この範囲内であるとジオメトリの 2 つのポイントまたは部分が同一であると見なされる距離を定義します。許容度は、ペン先の太いマーカを使用して描画したポイントと線で表現されているすべてのジオメトリにおける、ペン先の太さであると考えられます。この太いマーカを使用して描画した場合に接触するすべての部分が許容度内にあると見なされます。2 つのポイントが許容度からまったく同じ距離分離れている場合、これらのポイントは許容度内とは見なされません。

許容度の単純な例として、たとえば、許容度が 0.5 の場合には、Point( 14.2, 28.4 ) と Point( 14.4, 28.2 ) は等しいと見なされます。これは、(X と Y が同じ単位で表される) 2 点間の距離が約 0.283 であり、許容度よりも小さいためです。通常、許容度にはこの単純な例よりずっと小さな値が設定されます。

極端に小さいジオメトリは、許容度により無効になる場合があります。長さが許容度より短い線は無効です (ポイントが等しいため)。また、すべてのポイントが許容度内にある同様の多角形は無効と見なされます。

グリッドにスナップと許容度は空間参照系に設定され、X と Y (または経度の緯度) の座標には常に同じ単位が使用されます。「グリッドにスナップ」と許容度は、厳密でない算術データや不正確なデータに関する問題を解消するために一緒に使用されます。ただし、これらの動作が空間操作の結果に及ぼす影響について注意してください。

---

**注意：**平面空間参照系の場合、空間操作が不正な結果になる場合があるため、グリッドサイズを 0 に設定することはおすすめしません。曲面空間参照系の場合、グリッドサイズと許容度を 0 に設定してください。SAP Sybase IQ は、曲面操作を実行する場合、固定されたグリッドサイズと許容度を内部投影で使用します。

---

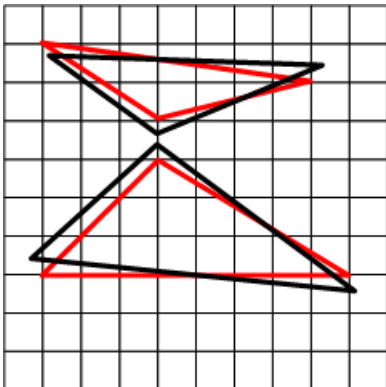
次の例は、グリッドサイズと許容度の設定が空間の計算に与える影響を示しています。

#### 例 1：「グリッドにスナップ」が交差の結果に与える影響

2 つの三角形 (黒で表示) が空間参照系にロードされます。ここで、許容度はグリッドサイズと同じに設定され、図のグリッドはグリッドサイズに基づいています。黒の三角形の頂点をグリッドにスナップした後の三角形を、赤い三角形で表しています。元の三角形 (黒) はそれぞれの許容度の範囲内に適切に収まっていますが、スナップされたバージョンの赤の三角形は許容度内にないことに注意してください。ST\_Intersects は、これらの 2 つのジオメトリに対して 0 を返します。許

## 空間に関する高度なトピック

容量がグリッドサイズより大きい場合、ST\_Intersects はこれらの2つのジオメトリに対して 1 を返します。

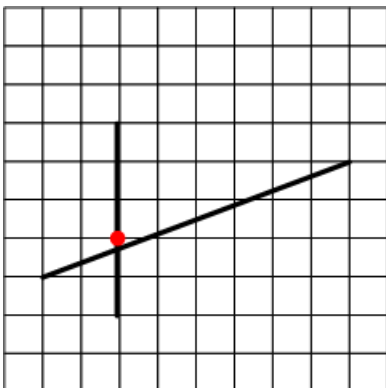


### 例 2：許容度が交差の結果に与える影響

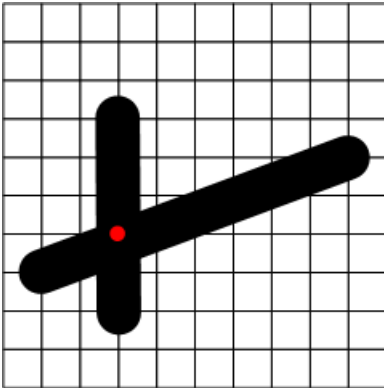
次の例では、許容度が 0 に設定された空間参照系内に 2 つの線があります。2 つの線の交差点は、グリッドの最も近い頂点にスナップされています。許容度が 0 に設定されているため、2 つの線の交差点が斜めの線と交差しているかどうかを検査するテストでは false が返されます。

つまり、許容度が 0 の場合、次の式は 0 を返します。

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

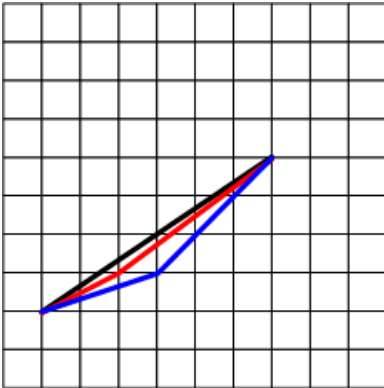


許容度をグリッドサイズと同じに設定すると (デフォルト)、交差点が太い斜めの線の内側に収まります。したがって、交差点が斜めの線と許容度内で交差するかどうかのテストはパスします。



### 例 3：許容度と推移性

許容度が使用されている場合の空間の計算では、推移性が保持される必要はありません。たとえば、許容度がグリッドサイズと等しい空間参照系に、次の3つの線があるとします。



ST\_Equals メソッドでは、黒と赤の線および赤と青の線はそれぞれ許容度内にあると見なされますが、黒と青の線は許容度内にあるとは見なされません。ST\_Equals は推移的ではありません。

ST\_OrderingEquals はこれらの各線を異なると見なし、推移的です。

### 例 4：グリッドと許容度の設定が不正確なデータに与える影響

投影平面空間参照系にあるデータの精度が、10 cm 以内でほぼ正確、10 m 以内では必ず正確であるとして、この場合3つの選択肢があります。

1. SAP Sybase IQ によって選択されるデフォルトのグリッドサイズと許容度を使用します。これは、通常は、使用するデータの精度よりも高い精度になります。これにより最大精度が提供されますが、ST\_Intersects、ST\_Touches、ST\_Equals などの述部は、ジオメトリ値の精度に応じて、一部のジオメトリに

## 空間に関する高度なトピック

対して予期したものと異なる結果を返すことがあります。たとえば、1つの境界を共有する隣接した2つの多角形において、一番右側の多角形の左端から数メートルのところに一番左側の多角形の境界データがある場合、`ST_Intersect` は `true` を返さない可能性があります。

2. グリッドサイズを、最も精度の高いデータを表すことができるくらい小さくし (この場合、10 cm)、許容度の4分の1以下に設定します。許容度は、データが常にどの程度の精度で距離を表すかを示す値に設定します (この場合、10 m)。この方式では、精度を失うことなくデータが格納され、データが10 m以内でしか正確でない場合でも述部が予期した結果が返されます。
3. グリッドサイズと許容度をデータの精度と同じに設定します (この場合、10 m)。この方法では、データはその精度内にスナップされますが、10 mより精度が高いデータの場合は精度が失われます。  
多くの場合、述部は予期した結果を返しますが、そうでない場合もあります。たとえば、2つのポイントが10 cm以内にあるが、グリッドの交差の中間点近くにある場合、2つのポイントはそれぞれ別の方向にスナップされて、ポイント間の距離が10 mになってしまいます。このため、グリッドサイズと許容度をデータの精度と同じに設定することは、この場合おすすめしません。

## 補間が空間の計算に与える影響

補間とは、ジオメトリ中の既知のポイントを使用して、未知のポイントを概算するプロセスです。いくつかの空間メソッドおよび述部では、円弧が関係する計算を実行するときに、補間が使用されます。補間によって、円弧は一連の直線に置き換えられます。たとえば、4分円を表す円ストリングを補間して、コントロールポイントが11ある線ストリングに変換する場合があります。

### 補間の例

1. Interactive SQL で、サンプルのデータベースに接続し、次の文を実行して円ストリングを格納する `arc` という変数を作成します。

```
CREATE VARIABLE arc ST_CircularString;
```

2. 次の文を実行して、円ストリングを作成し、それを `arc` 変数に格納します。

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

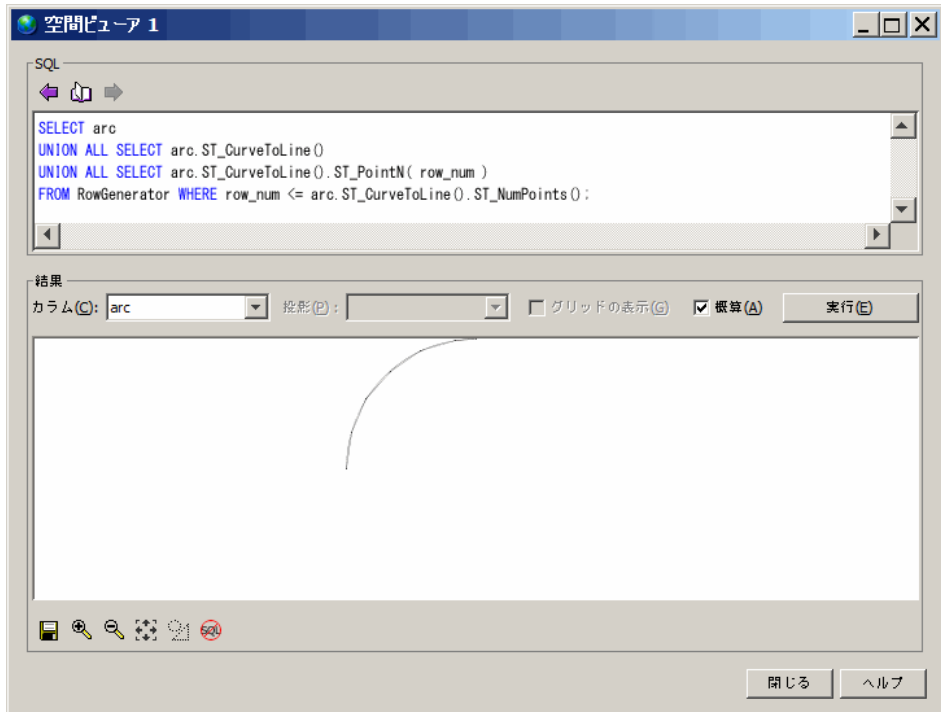
3. 次の文を実行して、`st_geometry_interpolation` オプションを使用し、一時的に相対許容度を1%に設定します。

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

相対許容度を1%に設定するのはオプションですが、この例では、補間による影響をよりわかりやすくするために使用します。

4. [空間ビューア]を開き (Interactive SQL で [ツール] » [空間ビューア] を選択)、次のクエリを実行して円ストリングを表示します。

```
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();
```



円弧が一連の線ストリングに分解されていることに注意してください。相対許容度が 1% に設定されているため、各直線セグメントは実際の円弧の内側にずれて表示されます。補間された線ストリングと実際の円弧との間の最大距離は、円弧の半径の 1% になります。

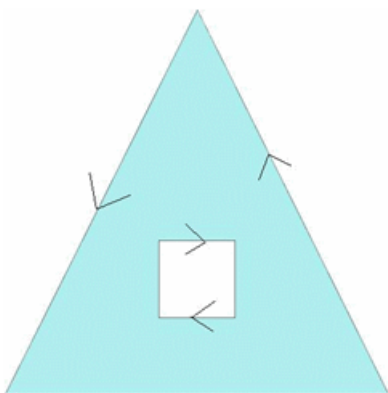
## 多角形リングの方向操作

SAP Sybase IQ では、まず、多角形を構成するリングの方向によって多角形が解釈されます。定義された点の順序でリングを移動した場合、多角形の内側がリングの左側になります。平面と曲面の空間参照系では、同じルールが適用されます。ほとんどの場合、外部リングは反時計回りの方向で、内部リングは逆の方向(時計回り)となります。例外は、ROUND EARTH に北極または南極が含まれるリングの場合です。

デフォルトでは、多角形が SAP Sybase IQ の内部的なリング方向と異なるリング方向で作成されている場合は、自動的に再方向付けされます。CREATE SPATIAL REFERENCE SYSTEM 文の POLYGON FORMAT 句を使用して、入力データの多角形リングの方向を指定します。これは、空間参照系の入力データで同じリングの方向を使用している場合にのみ実行する必要があります。POLYGONFORMAT は、一部の多角形および複数面のコンストラクタでも指定できます。

たとえば、多角形を作成し、ポイントを時計回りの順で指定すると (Polygon((0 0, 5 10, 10 0, 0 0), (4 2, 4 4, 6 4, 6 2, 4 2))), データベースサーバは、ポイントが反時計回りになるように自動的に並べ替えます ()。

```
Polygon((0 0, 10 0, 5 10, 0 0), (4 2, 4 4, 6 4, 6 2, 4 2)).
```



内部リングが外部リングより前に指定された場合、外部リングが最初のリングとして表示されます。

曲面空間参照系で多角形の再方向付けが機能するためには、多角形の直径は 160 度に制限されます。

## ジオメトリの内部、外部、境界の操作

---

ジオメトリの**内部**とは、ジオメトリの一部であるすべてのポイントです (境界を除く)。

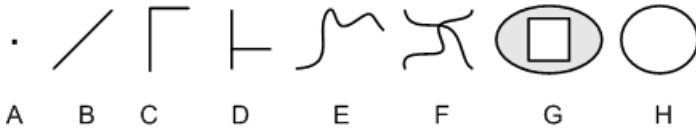
ジオメトリの**外部**とは、ジオメトリの部分ではないすべてのポイントです。これには、内部リングの内側の空間も含まれます。たとえば、多角形に穴の開いている場合です。同様に、線ストリングのリングの内側と外側の空間は外部と見なされます。

ジオメトリの**境界**とは、ST\_Boundary メソッドによって返される内容です。

ジオメトリの境界について知っていると、別のジオメトリと比較して、2つのジオメトリの関連を判別するときに役立ちます。ただし、すべてのジオメトリには

内部と外部がありますが、すべてのジオメトリに境界があるわけではありません。また、その境界は必ずしも直感的ではありません。

境界が直感的ではないジオメトリのケースを次に示します。



- **ポイント** – ポイント (たとえば A) には境界はありません。
- **線と曲線** – 線と曲線 (B、C、D、E、F) の境界は終了ポイントです。ジオメトリ B、C、E には、境界として2つの終了ポイントがあります。ジオメトリ D には、境界として4つの終了ポイントがあり、ジオメトリ F にも4つあります。
- **多角形** – 多角形 (たとえば G) の境界は、その外部リングと内部リングです。
- **リング** – リングとは、開始ポイントが終了ポイントと同じで、交差することがない曲線 (たとえば H) であり、境界はありません。

## 空間の比較操作

ジオメトリが別のジオメトリと等しいかどうかをテストするために使用できるメソッドには、`ST_Equals` と `ST_OrderingEquals` の2つがあります。これらのメソッドで実行される比較と返される結果は異なっています。

- **ST\_Equals** – ポイントが指定される順序は関係ありません。ポイントの比較では許容度が考慮されます。許容度内で同じ空間を占有している場合は、ジオメトリについても等しいと見なされます。たとえば、2つの線ストリングが同じ空間を占有しており、片方にはより多くのポイントが定義されている場合でも、2つの線ストリングが等しいと見なされることを意味します。
- **ST\_OrderingEquals** – `ST_OrderingEquals` では、2つのジオメトリには同じオブジェクト階層が含まれ、その階層には `ST_OrderingEquals` で等しいと見なされる順序でまったく同じポイントが存在している必要があります。つまり、2つのジオメトリがまったく同一である必要があります。

`ST_Equals` と `ST_OrderingEquals` を使用して比較を実行したときの結果の差異を確認するために、次の線を比較してみてください。`ST_Equals` では、これらのすべての線が等しいと見なされます (線 C が許容度内であることが前提)。ただし、`ST_OrderingEquals` では、これらのすべての線は等しいとは見なされません。

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

### SAP Sybase IQ によるジオメトリ比較の実行方法

データベースサーバは、ST\_OrderingEquals を使用して、GROUP BY、DISTINCT などの操作を実行します。

たとえば、次のクエリを処理するとき、2つの shape 式で ST\_OrderingEquals() = 1 である場合、サーバは2つのローを等しいと見なします。

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

SQL文では、等しい(=)または等しくない(<>)または(!=)の演算子を使用して2つのジオメトリを比較できます。サブクエリやANYまたはALLキーワードを含む検索条件も使用できます。ジオメトリはIN検索条件でも使用できます。たとえば geom1 IN (geom-expr1, geom-expr2, geom-expr3) のように指定します。これらすべての検索条件では、等価性は ST\_OrderingEquals セマンティックを使用して評価されます。

その他の比較演算子を使用して、ジオメトリが別のジオメトリより小さいかどうか、または大きいかどうかを判別することはできません(たとえば、geom1 < geom2 は受け入れられません)。これは、ジオメトリ式を ORDER BY 句に含めることはできないことを意味します。ただし、集合に含まれているかどうかはテストできます。

## 空間の関係操作

---

最良のパフォーマンスを得るためには、ST\_Within または ST\_Touches などのメソッドを使用して、ジオメトリ間の単一の特定の関係をテストします。ただし、複数の関係をテストする場合、一度に複数の関係をテストできる ST\_Relate メソッドの方が適しています。ST\_Relate は、述部の異なる解釈をテストする場合にも役に立ちます。

ST\_Relate の最も一般的な使用法は、テストする関係を厳密に指定して、述部として使用する方法です。ただし、ST\_Relate を使用して、2つのジオメトリ間で可能なすべての関係を判別することもできます。

### 述部としての ST\_Relate の使用

ST\_Relate は、内部、境界、外部の**交差テスト**を実行することによって、ジオメトリ間の関係を評価します。ジオメトリ間の関係は、DE-9IM (Dimensionally Extended 9 Intersection Model) フォーマットの9文字の文字列で記述されます。この文字列の各文字は、交差テストの結果の次元を表します。

ST\_Relate を述部として使用する場合、テストする交差の結果を示した DE-9IM 文字列を渡します。指定した DE-9IM 文字列の条件をジオメトリが満たしている場



合、ST\_Relate は **1** を返します。条件が満たされない場合は、**0** を返します。片方のジオメトリ、または両方が NULL の場合、**NULL** を返します。

9 文字の DE-9IM 文字列は、内部、境界、外部間の交差テストのペアごとのマトリックスをフラットにした表現です。次の表に、実行される順序(左から右、上から下)で9つの交差テストを示します。

	g2 内部	g2 境界	g2 外部
g1 内部	Interior (g1) $\cap$ Interior (g2)	Interior (g1) $\cap$ Boundary (g2)	Interior (g1) $\cap$ Exterior (g2)
g1 境界	Boundary (g1) $\cap$ Interior (g2)	Boundary (g1) $\cap$ Boundary (g2)	Boundary (g1) $\cap$ Exterior (g2)
g1 外部	Exterior (g1) $\cap$ Interior (g2)	Exterior (g1) $\cap$ Boundary (g2)	Exterior (g1) $\cap$ Exterior (g2)

DE-9IM 文字列を指定する場合、9 文字の各文字に \*、0、1、2、T、または F を指定できます。これらの値は、交差によって作成されるジオメトリの次元数を表しています。

指定する文字	交差テストが返す結果
T	次のいずれか：0、1、2 (任意の次元の交差)
F	-1
*	-1、0、1、2 (任意の値)
0	0
1	1
2	2

ST\_Relate と Within 述部用のカスタム DE-9IM 文字列を使用して、ジオメトリが別のジオメトリ内にあるかどうかをテストするとします。

```
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***' );
```

これは、交差テストを実行するときに、ST\_Relate に次の条件で検索するように問い合わせるのと同じです。

	g2 内部	g2 境界	g2 外部
g1 内部	次のいずれか： <b>0、1、2</b>	次のいずれか：0、1、2、-1	<b>-1</b>

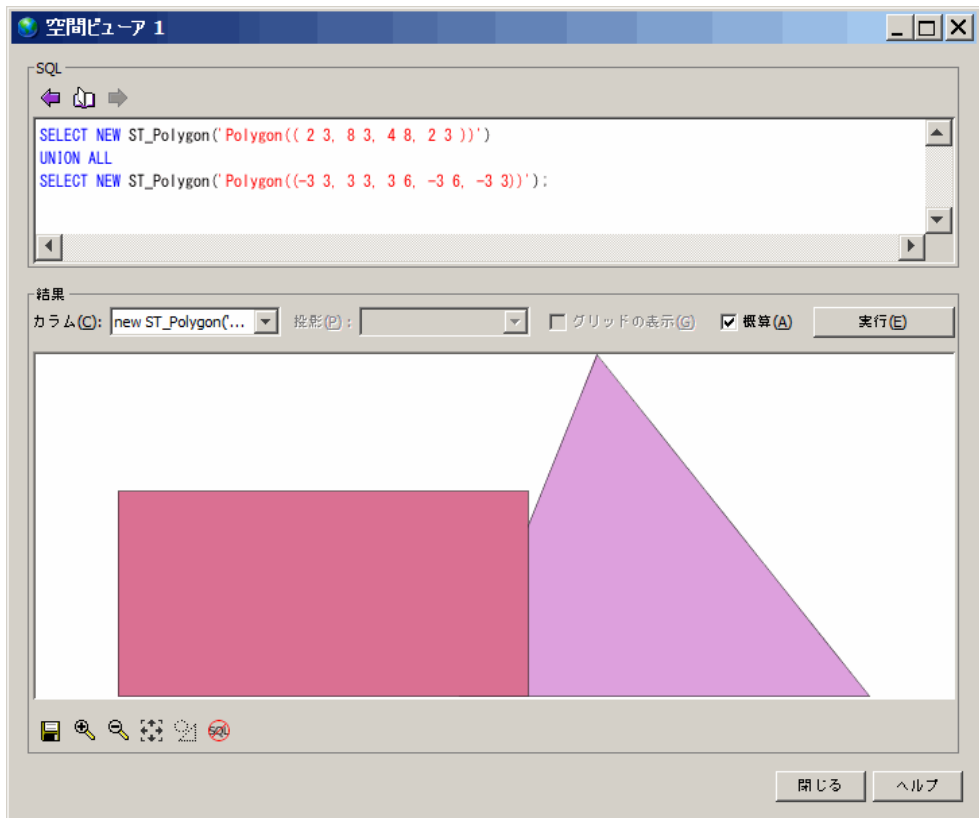
## 空間に関する高度なトピック

g1 境界	次のいずれか：0、1、2、-1	次のいずれか：0、1、2、-1	<b>-1</b>
g1 外部	次のいずれか：0、1、2、-1	次のいずれか：0、1、2、-1	次のいずれか：0、1、2、-1

クエリを実行すると、ST\_Relate は最初のジオメトリが2番目のジオメトリ内でないことを示す0を返します。

2つのジオメトリを表示して、その形状をテストしている内容と比較するには、次の文を **Interactive SQL** の空間ビューア ([ツール]»[空間ビューア]) で実行します。

```
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))');
```



### 述部以外での ST\_Relate の使用

述部以外で ST\_Relate を使用すると、2つのジオメトリ間のすべての関係が返されます。

たとえば、前の例で使用したものと同じ2つのジオメトリがあり、それらの関係を知りたいとします。次の文を **Interactive SQL** で実行すると、ジオメトリ間の関係を定義した **DE-9IM** 文字列が返されます。

```
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** は **DE-9IM** 文字列 **212111212** を返します。

この値をマトリックス表示すると、多くの交差ポイントがあることがわかります。

	g2 内部	g2 境界	g2 外部
g1 内部	2	1	2
g1 境界	1	1	1
g1 外部	2	1	2

## 空間の次元の操作

各ジオメトリサブクラスは、独自のプロパティを持っている以外に、**ST\_Geometry** スーパータイプからプロパティを継承しています。ジオメトリサブタイプは、次のいずれかの次元値を持っています。

- **-1** - 値 **-1** は、ジオメトリが空であることを示します (ポイントが1つもない)。
- **0** - 値 **0** は、ジオメトリが長さまたは面積を持たないことを示します。サブタイプ **ST\_Point** と **ST\_MultiPoint** は次元値 **0** を持ちます。1つのポイントは、座標の単一のペアによって表すことができるジオメトリ特性を表します。接続されていないポイントのクラスは **MultiPoint** 特性を表します。
- **1** - 値 **1** は、ジオメトリに長さがあるが、面積がないことを示します。次元 **1** を持つ一連のサブタイプは、**ST\_Curve** のサブタイプ (**ST\_LineString**、**ST\_CircularString**、および **ST\_CompoundCurve**)、またはこれらのタイプを含んでいるが面を持たないコレクションタイプです。GIS データでは、これらの次元 **1** のジオメトリは、線形特性 (河川、水系、道路網など) を定義するために使用されます。
- **2** - 値 **2** は、ジオメトリが面積を持っていることを示します。次元 **2** を持つ一連のサブタイプは、**ST\_Surface** のサブタイプ (**ST\_Polygon** と **ST\_CurvePolygon**)、またはこれらのタイプを含んでいるコレクションタイプです。多角形と複数多角形は、定義された面積を囲む外周を持つジオメトリ特性 (湖、公園など) を表します。

ジオメトリの次元は、ジオメトリの各ポイントの座標次元の数とは関係がありません。

## 空間に関する高度なトピック

1つの `ST_GeomCollection` には、次元の異なるジオメトリを含めることができ、最も高い次元のジオメトリが返されます。

# チュートリアル：空間機能の実験

このチュートリアルでは、SAP Sybase IQ のいくつかの空間機能を実験します。これを行うには、まず ESRI シェイプファイルをサンプルデータベース (demo.db) にロードして、実験のための有効な空間データを用意します。

このチュートリアルは次に示す部分に分かれています。

## 権限

このチュートリアルを実行するには、次の権限が必要です。

- MANAGE ANY SPATIAL OBJECT システム権限
- CREATE TABLE システム権限
- WRITE FILE システム権限
- GROUPO.SpatialContacts テーブルに関する SELECT 権限

## レッスン 1：追加の測定単位と空間参照系のインストール

---

このレッスンでは、sa\_install\_feature システムプロシージャを使用し、このチュートリアルの後半で必要となる多くの定義済みの測定単位と空間参照系をインストールする方法について説明します。

### 前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロールおよび権限を持っていることを前提としています。「チュートリアル：空間機能の実験」

### 手順

1. InteractiveSQL を使用して、サンプルデータベース (demo.db) を起動して接続します。

サンプルデータベースは %IQDIRSAMP16% にあります。

2. Interactive SQL で次の文を実行します。

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

文が完了すると、追加の測定単位と空間参照系がインストールされています。

3. データベースにインストールされている測定単位を判別するには、次のクエリを実行します。

```
SELECT * FROM ST_UNITS_OF_MEASURE;
```

4. データベースにインストールされている空間参照系を判別するには、Interactive SQL で次のクエリを実行します。

```
SELECT * FROM ST_SPATIAL_REFERENCE_SYSTEMS;
```

インストールされた空間参照系のリストが返されます。

## レッスン 2：ESRI シェイプファイルデータのダウンロード

---

### 前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン 1：追加の測定単位と空間参照系のインストール」を参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロールおよび権限を持っていることを前提としています。「チュートリアル：空間機能の実験」

### 手順

1. `c:\¥temp¥massdata` という名前のローカルディレクトリを作成します。
2. 次の URL に移動します：<http://www2.census.gov/cgi-bin/shapefiles2009/national-files>
3. ページの右側の [State- and County-based Shapefiles] ドロップダウンから [Massachusetts] をクリックして、[submit] をクリックします。
4. ページの左側で、[5-Digit ZIP Code Tabulation Area (2002)] をクリックして、[Download Selected Files] をクリックします。
5. プロンプトが表示されたら、zip ファイル `multiple_tiger_files.zip` を `c:\¥temp¥massdata` に保存し、ファイルの内容を解凍します。これにより、`25_MASSACHUSETTS` という名前のサブフォルダが作成され、`t1_2009_25_zcta5.zip` という名前の別の zip ファイルがそこに作成されます。
6. `t1_2009_25_zcta5.zip` の内容を `C:\¥temp¥massdata` に解凍します。

これにより、空間データをデータベースにロードするために使用できる ESRI シェイプファイル (.shp) を含む 5 つのファイルが解凍されます。

## レッスン 3：ESRI シェイプファイルデータのロード

このレッスンでは、ESRI シェイプファイル内のカラムを特定し、その情報を使用して、データをロードするテーブルを作成する方法を説明します。

### 前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン 1：追加の測定単位と空間参照系のインストール」を参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロールおよび権限を持っていることを前提としています。「チュートリアル：空間機能の実験」

### 手順

権限に問題があるために、いずれかの手順を実行できない場合は、-gl データベースオプションの設定値を管理者に問い合わせしてから、st\_geometry\_load\_shapefile システムプロシージャの権限に関するセクションを参照して必要な権限を判別してください。

1. 空間データは特定の空間参照系に関連付けられているため、データベースにデータをロードする場合、同じ空間参照系、または少なくとも等価な定義を持つ空間参照系にロードする必要があります。ESRI シェイプファイルの空間参照系情報を確認するには、プロジェクトファイル c:\%temp%\massdata\%t1\_2009\_25\_zcta5.prj をテキストエディタで開きます。このファイルには、この手順に必要な空間参照系情報が含まれています。

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

文字列 `GCS_North_American_1983` は、データに関連付けられている空間参照系の名前です。

2. `ST_SPATIAL_REFERENCE_SYSTEMS` ビューに問い合わせると (`SELECT * FROM ST_SPATIAL_REFERENCE_SYSTEMS WHERE srs_name='GCS_North_American_1983';`); この名前は定義済みの SRS のリストにはありません。ただし、同じ定義の空間参照系を問い合わせ、代わりにそれを使用できます。

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

このクエリは、1つの同じ定義を持つ SRID 4269 の空間参照系 NAD83 を返します。これがシェイプファイルからデータをロードするために割り当てる SRID です。

- Interactive SQL で、次の文を実行して、Massdata というテーブルを作成し、シェイプファイルをテーブルにロードして、データに SRID 4269 を割り当てます。ロードされるまで1分間かかることがあります。

```
CALL st_geometry_load_shapefile ( 'c:\temp\massdata\
\t1_2009_25_zcta5.shp',
4269,
'Massdata' );
```

**注意：**[インポートウィザード]でも、シェイプファイルからのデータのロードがサポートされています。

- Interactive SQL で、テーブルをクエリし、シェイプファイルにあったデータを表示します。

```
SELECT * FROM Massdata;
```

結果の各ローは、郵便番号区域のデータを表します。

geometry カラムには、多角形(1つの領域)または複数多角形(複数の隣接しない領域)として、郵便番号区域のシェイプ情報が保持されています。

- ZCTA5CE カラムには郵便番号が格納されています。このカラムをチュートリアルの後半で簡単に参照できるようにするため、Interactive SQL で ALTER TABLE 文を実行して、カラム名を ZIP に変更します。:

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

- 2つのカラム INTPTLON と INTPTLAT は、郵便番号区域の中心ポイントの X 座標と Y 座標を表しています。Interactive SQL で次の ALTER TABLE 文を実行して、ST\_Point タイプの CenterPoint というカラムを作成し、それぞれの X と Y を CenterPoint の値にします。

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

これで、Massdata.CenterPoint の ST\_Point 値が、Massdata.geometry に格納された郵便番号区域の中心ポイントを表すようになりました。

- 個別のジオメトリ(郵便番号区域)をシェイプとして表示するには、Massdata.geometry の最初の値以外をダブルクリックし、[カラムの値] ウィンドウの [空間プレビュー] タブをクリックします。

値が大きすぎるというエラー、または結果にプライマリキーを含めることを推奨するというエラーを受け取った場合、Interactive SQL で表示するために値が

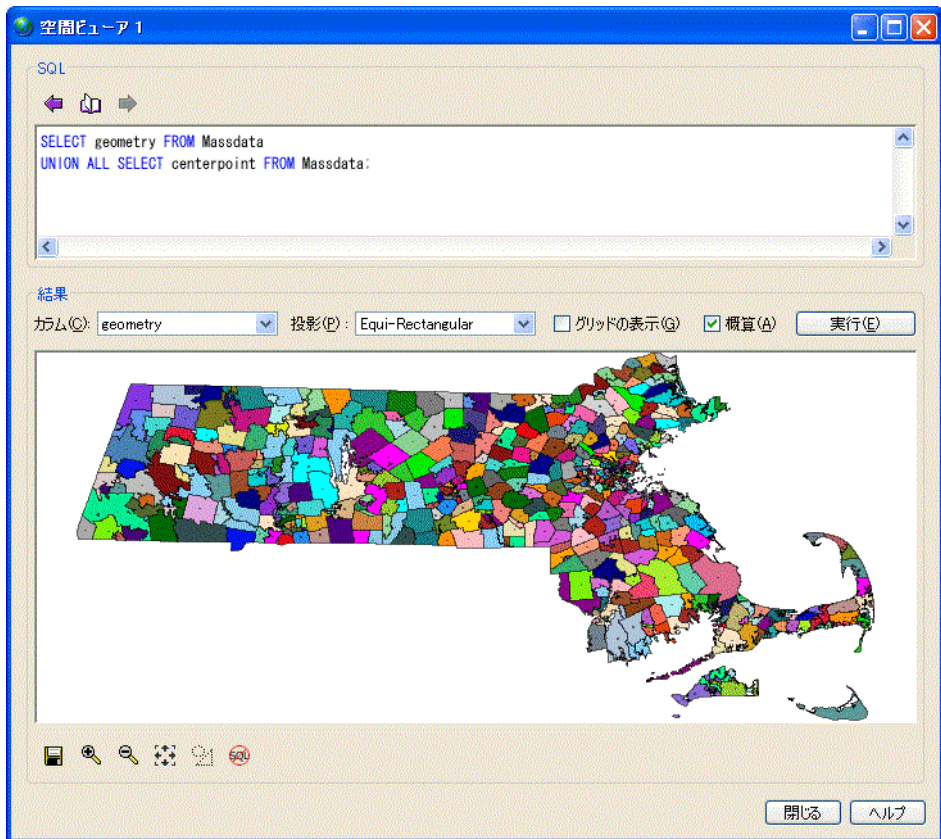


トランケートされたことが原因です。これを解決するには、クエリを変更して、結果にプライマリーキーが含まれるようにするか、Interactive SQL の [トランケーションの長さ] 設定を調整します。Interactive SQL でジオメトリを表示するために、ジオメトリを問い合わせるたびにプライマリーキーを含める必要をなくしたい場合は、設定の変更をおすすめします。

Interactive SQL の [トランケーションの長さ] 設定を変更するには、[ツール] » [オプション] » [SAP Sybase IQ] をクリックし、[トランケーションの長さ] を 100000 などの大きい値に設定します。

8. データセット全体を 1 つのシェイプとして表示するには、[ツール] » [空間ビューア] をクリックして、SAP Sybase IQ の [空間ビューア] を開き、Interactive SQL で次のクエリを実行します。

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



ESRI シェイプファイルデータがロードされます。

## レッスン 4：空間データのクエリ

---

このレッスンでは、いくつかの空間メソッドを使用して、意味のあるコンテキストでデータを問い合わせる方法について説明します。また、距離を計算する方法も学習するためデータベースに測定単位を追加する必要があります。

### 前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1：追加の測定単位と空間参照系のインストール」を参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロールおよび権限を持っていることを前提としています。「チュートリアル：空間機能の実験」

### 手順

SpatialContacts テーブルと Massdata テーブルのどちらか、または両方でクエリを実行します。データベースにすでに存在している SpatialContacts には、多くは Massachusetts に在住する人々の名前と連絡先情報が格納されています。

1. Interactive SQL で、郵便番号区域 01775 に関連付けられるジオメトリを保持するために、@Mass\_01775 という変数を作成します。

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

2. 郵便番号区域 01775 と周辺の郵便番号区域のすべての連絡先を SpatialContacts から検索するとします。このためには、交差するジオメトリ、または指定したジオメトリと同じジオメトリを返す、ST\_Intersects メソッドを使用します。

Interactive SQL で次の文を実行します。

```
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 のすべてのローは、同じ空間参照系 (SRID 4269) に関連付けられています。これは、geometry カラムを作成するときに SRID 4269 を割り当て、そのカラムにデータをロードしたためです。

ただし、**宣言されていない** ST\_Geometry カラム (つまり、SRID が割り当てられていない) を作成することもできます。これは、異なる SRS が関連付けられて

いる空間値を単一のカラムに格納したい場合に必要になります。これらの値に対して操作が実行されると、各値に関連付けられている空間参照系が使用されます。

宣言されていないカラムを使用する場合のリスクの1つは、データベースサーバでは、宣言されていないカラムのデータに関連付けられている空間参照系の変更が防止されないことです。

カラムに宣言された SRID がある場合、データベースサーバはデータに関連付けられている空間参照系を変更することを許可しません。最初に、宣言されたカラムのデータをアンロードしてトランケートしてから、空間参照系を変更し、データを再ロードする必要があります。

ST\_SRID メソッドを使用すると、宣言されているかどうかに関係なく、カラムの値に関連付けられている SRID を判別できます。たとえば、次の文は Massdata.geometry カラムの各ローに割り当てられている SRID を表示します。

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

- ST\_CoveredBy メソッドを使用すると、ジオメトリが別のジオメトリ内に完全に含まれているかどうかをチェックできます。たとえば、Massdata.CenterPoint (ST\_Point タイプ) には郵便番号区域の中心の緯度/経度の座標が含まれ、Massdata.geometry には郵便番号区域を反映した多角形が含まれます。Interactive SQL で次のクエリを実行することによって、郵便番号区域外に設定されている CenterPoint 値がないことを簡単に確認できます。

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

ローが1つも返されないため、すべての CenterPoint 値が Massdata.geometry の関連付けられているジオメトリ内にあることを示しています。このチェックは CenterPoint 値が本当に中心にあるかどうかは検証していません。そのためには、データを平面空間参照系に投影し、ST\_Centroid メソッドを使用して CenterPoint 値をチェックする必要があります。投影についてはこのチュートリアルの後半で学習します。

- ST\_Distance メソッドを使用すると、郵便番号区域の中心点間の距離を測定できます。たとえば、郵便番号区域 01775 から 100 マイル以内の郵便番号をリストしたいとします。この処理は、Interactive SQL で次のクエリを実行します。

```
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. 正確な距離を知ることが重要ではない場合、代わりに `ST_WithinDistance` メソッドを使用してクエリを作成できます。これは、特定のデータセット (特に、大きなジオメトリ) に対してよりよいパフォーマンスが得られます。

```
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;
```

空間データに関するクエリが実行されます。

## レッスン 5：SVG への空間データの出力

このレッスンでは、SVG ドキュメントを作成し、WKT で表現された複数多角形を表示します。ジオメトリを SVG フォーマットにエクスポートすると、Interactive SQL または SVG に互換性のあるアプリケーションで表示できます。

### 前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン 1：追加の測定単位と空間参照系のインストール」を参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロールおよび権限を持っていることを前提としています。「チュートリアル：空間機能の実験」

### 手順

1. Interactive SQL で次の文を実行して、ジオメトリの例が設定された変数を作成します。

```
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. Interactive SQL で、次の SELECT 文を実行して、`ST_AsSVG` メソッドを呼び出します。

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

結果セットには単一のローがあり、SVG イメージが含まれています。このイメージは、Interactive SQL の [SVG プレビュー] 機能を使用して表示できます。これを行うには、結果ローをダブルクリックして、[SVG プレビュー] タブを選択します。正方形のジオメトリが別の正方形のジオメトリ内に表示されます。

**注意：** デフォルトでは、Interactive SQL は [結果] ウィンドウ枠内の値を 256 文字にトランケートします。Interactive SQL が完全なカラム値を読み込めないことを示すエラーを返す場合、トランケーション値を増やします。これを行うには、[ツール] » [オプション] をクリックし、左ウィンドウ枠で [SAP Sybase IQ] をクリックします。[結果] タブで、[トランケーションの長さ] を 5000 などの大きい値に変更します。[OK] をクリックして変更を保存し、再びクエリを実行してから、再度そのローをダブルクリックします。

3. 前の手順では、SVG イメージを Interactive SQL 内でプレビューする方法を説明しました。ただし、結果の SVG をファイルに書き込んで、外部のアプリケーションで読み込めるようにするとより便利です。xp\_write\_file システムプロシージャまたは WRITE\_CLIENT\_FILE 関数 [文字列] を使用すると、データベースサーバまたはクライアントコンピュータの相対位置にあるファイルに書き込むことができます。この例では、OUTPUT 文 [Interactive SQL] を使用します。

Interactive SQL で、次の SELECT 文を実行して ST\_AsSVG メソッドを呼び出し、ジオメトリを myPolygon.svg という名前のファイルに出力します。

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

この文には、QUOTE '' と ESCAPES OFF 句を含める必要があります。そうしないと、ホワイトスペースを維持するために改行復帰文字と一重引用符が XML に挿入され、出力が無効な SVG ファイルになります。

4. Web ブラウザまたは SVG イメージの表示をサポートするアプリケーションで SVG を開きます。または、SVG をテキストエディタで開くと、ジオメトリの XML を表示できます。
5. ST\_AsSVG メソッドは、単一のジオメトリから SVG イメージを生成します。場合によっては、グループ内のすべてのシェイプが含まれる SVG イメージを生成することがあります。ST\_AsSVGAggr メソッドは、複数のジオメトリを単一の SVG イメージに結合する集合関数です。まず、Interactive SQL を使用して、SVG イメージを保持する変数を作成し、ST\_AsSVGAggr メソッドを使用してそれを生成します。

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

@svg 変数は、現在、Massdata テーブル内のすべての郵便番号区域を表す SVG イメージを保持しています。'attribute=fill="black"' は、生成されたイメージに使用される塗りつぶしの色を指定します。指定しない場合、データ

ベースサーバは任意の塗りつぶしの色を選択します。目的の SVG イメージを含んだ変数が作成されたので、それをファイルに書き込んで別のアプリケーションで表示できます。Interactive SQL で次の文を実行して、SVG イメージをデータベースサーバの相対位置にあるファイルに書き込みます。

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

WRITE\_CLIENT\_FILE 関数も、クライアントアプリケーションの相対位置にあるファイルに書き込むために使用できますが、適切な権限を有効にするために追加の手順が必要となります。SVG データをサポートするアプリケーションで SVG イメージを開くと、次のようなイメージが表示されます。



このイメージは一様に黒ではありません。隣接する郵便番号区域の境界の間に小さなギャップがあります。ジオメトリ間には実際に白線があり、これは SVG が描画される方式の特徴です。現実にはデータにギャップはありません。太くて白い線は河川と湖です。

ジオメトリは SVG として表示されています。

## レッスン 6：空間データの投影

---

このレッスンでは、面積の計算と距離の測定ができるように、平面モデルを使用する空間参照系にデータを投影する方法を説明します。

### 前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン 1：追加の測定単位と空間参照系のインストール」を参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロールおよび権限を持っていることを前提としています。「チュートリアル：空間機能の実験」

## 手順

Massdata の空間値は、ESRI シェイプファイルからデータベースにデータをロードするときに、SRID 4269 (NAD83 空間参照系) が割り当てられています。SRID 4269 は曲面の空間参照系です。ただし、ジオメトリの面積やいくつかの空間述部などの計算は、曲面モデルではサポートされません。現在、データが曲面空間参照系に関連付けられている場合、平面空間参照系に値を投影する新しい空間カラムを作成し、そのカラムに対して計算を実行できます。

1. 郵便番号区域を表す多角形の面積を測定するには、Massdata.geometry のデータを平面空間参照系に投影する必要があります。

Massdata.geometry のデータを投影する適切な SRID を選択するには、Interactive SQL を使用して、次のように単語 Massachusetts を含む SRID を ST\_SPATIAL\_REFERENCE\_SYSTEMS 統合ビューに問い合わせます。

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

これにより、Massachusetts のデータに使用するのに適したいくつかの SRID が返されます。このチュートリアルの用途には、SRID 3586 を使用します。

2. 次に、ST\_Transform メソッドを使用してジオメトリを SRID 3586 に投影するカラム Massdata.proj\_geometry を作成する必要があります。削除するには、Interactive SQL で次の文を実行します。

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

3. Massdata.proj\_geometry を使用して面積を計算できます。たとえば、Interactive SQL で次の文を実行します。

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

**注意：** ST\_Area は曲面空間参照系ではサポートされません。ST\_Distance はサポートされますが、ポイントジオメトリ間のみです。

4. 別の空間参照系への投影の影響を距離の計算で確認するには、次のクエリを使用し、曲面モデル (より正確) および投影された平面モデルを使用して、郵便番号区域の中心点間の距離を計算します。このデータでは両方のモデルがかなり一致します。これは、選択された投影がこのデータセットに適しているためです。

## チュートリアル：空間機能の実験

```
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. 郵便番号区域 01775 に隣接する郵便番号区域を検索するとします。これを行うには、`ST_Touches` メソッドを使用します。`ST_Touches` メソッドは、ジオメトリを比較して、ジオメトリが別のジオメトリに重なり合わずに接触しているかどうかをチェックします。`ST_Touches` の結果には、郵便番号 01775 のローは含まれません (`ST_Intersects` メソッドとは異なります)。

```
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. `ST_UnionAggr` メソッドを使用すると、郵便番号区域のグループの結合を表すジオメトリが返されます。たとえば、隣接する郵便番号区域の論理和 (ただし 01775 を含まない) を反映したジオメトリを表示するとします。

Interactive SQL で、[ツール] » [空間ビューア] をクリックして、次のクエリを実行します。

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

結果をダブルクリックして表示します。

データベースから完全なカラムを読み込めないことを示すエラーを受け取った場合、Interactive SQL の [トランケーションの長さ] 設定の値を増やします。これを行うには、Interactive SQL で、[ツール] » [オプション] » [SAP Sybase IQ] をクリックして、[トランケーションの長さ] に大きな値を設定します。クエリを再び実行してジオメトリを表示します。

これでチュートリアルは完了しました。



## 空間データへのアクセスとそのデータの操作

この項では、空間データへのアクセス、そのデータの操作および分析に使用できるタイプ、メソッド、コンストラクタについて説明します。空間データ型は、データ型やクラスと同じように考えることができます。各空間データ型には、データにアクセスするために使用するメソッドとコンストラクタが関連付けられています。



# 付録：SQL 文

このマニュアルで使用されている SQL 文の参考資料です。

## CREATE SPATIAL REFERENCE SYSTEM 文

---

空間参照系を作成するか、置き換えます。

クイックリンク：

「パラメータ」 (56 ページ)

「例」 (61 ページ)

「使用法」 (62 ページ)

「標準」 (62 ページ)

「パーミッション」 (62 ページ)

### 構文

```

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

srs-attribute - (構文に戻る)
  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' }
```

## パラメータ

(先頭に戻る) (55 ページ)

- **OR REPLACE** – OR REPLACE を指定すると、空間参照系がデータベースにまだ存在しない場合は空間参照系が作成され、存在する場合は置き換えられます。使用中の空間参照系を置き換えようとする、エラーが返されます。OR REPLACE 句を指定しないでデータベースにすでに存在する空間参照系を置き換えようとしても、エラーが返されます。
- **IF NOT EXISTS** – CREATE SPATIAL REFERENCE IF NOT EXISTS を指定すると、指定された名前の空間参照系がデータベースにすでに存在するかどうかチェックされます。存在しない場合は、データベースサーバによって空間参照系が作成されます。存在する場合は、追加のアクションは行われず、エラーも返されません。
- **IDENTIFIED BY** – 空間参照系の SRID (*srs-id*)。組織で *organization-srs-id* を指定して空間参照系を定義した場合は、*srs-id* をその値に設定してください。

IDENTIFIED BY 句が指定されていない場合は、デフォルトにより、ORGANIZATION 句または DEFINITION 句で定義された *organization-srs-id* に SRID が設定されます。デフォルトの SRID として使用できる *organization-srs-id* がどちらの句にも指定されていない場合は、エラーが返されます。

空間参照系が周知の座標系に基づいておらず、異なる測地線解釈がされている場合は、1000000000 (10 億) に周知の値を加えた値に *srs-id* 値を設定します。たとえば、測地線の空間参照系 WGS 84 (ID 4326) の平面解釈の SRID は、1000004326 になります。

SAP Sybase IQ で提供される周知の座標系に基づいていない空間参照系には、SRID 0 を除き 2000000000 (20 億) 以上の SRID が付与されます。2000000000 ~ 2147483647 の範囲の SRID 値は SAP Sybase IQ で予約されているため、この範囲の SRID は作成しないでください。

OGC などの定義機関または他のベンダで予約されている SRID を選択する可能性を低減するため、0 ~ 32767 (EPSG で予約済み) または 2147483547 ~ 2147483647 の範囲の SRID は選択しないでください。

また、SRID は 32 ビットの符号付き整数として格納されるため、 $2^{31}-1$  または 2147483647 を超える数値は指定できません。

- **DEFINITION** – 座標系のデフォルトを設定または上書きします。DEFINITION 句以外の句で属性が設定されている場合は、DEFINITION 句の設定内容に関係なく、他の句で指定された値が優先されます。

*definition-string* は、SQL/MM と OGC で定義されるような、空間参照系の Well Known Text 構文の文字列です。たとえば、次のクエリは WGS 84 の定義を返します。

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

Interactive SQL で戻り値をダブルクリックすると、読みやすい形で値が表示されます。

DEFINITION 句が指定されている場合は、*definition-string* が解析され、属性のデフォルト値の選択に使用されます。たとえば、*definition-string* に、*organization-name* と *organization-srs-id* を定義する AUTHORITY 要素が含まれる場合があります。

*definition-string* のパラメータ値は、SQL 文の句で明示的に設定した値で上書きされます。たとえば、ORGANIZATION 句が指定されていると、*definition-string* の ORGANIZATION の値が上書きされます。

- **ORGANIZATION** – 新しい空間参照系の基となる空間参照系を作成した組織に関する情報。
- **TRANSFORM DEFINITION** – 空間参照系に使用する変換の説明。現時点では、PROJ.4 変換のみがサポートされています。変換定義は、空間参照系間でデータを変換するときに ST\_Transform メソッドで使用されます。一部の变換は、*transform-definition-string* が定義されていなくても使用できます。
- **LINEAR UNIT OF MEASURE** – 空間参照系の線形測定単位。指定する値は、ST\_UNITS\_OF\_MEASURE システムビューで定義された線形測定単位と一致している必要があります。

この句が指定されておらず、DEFINITION 句で定義されていない場合、デフォルトは METRE です。事前に定義された測定単位をデータベースに追加するには、sa\_install\_feature システムプロシージャを使用します。

カスタム測定単位をデータベースに追加するには、CREATE SPATIAL UNIT OF MEASURE 文を使用します。

---

**注意：** METRE と METER はいずれも有効なスペルですが、SQL/MM 標準に準拠している METRE を使用することをおすすめします。

---

- **ANGULAR UNIT OF MEASURE** – 空間参照系の角度測定単位。指定する値は、ST\_UNITS\_OF\_MEASURE システムテーブルで定義された角度測定単位と一致している必要があります。

この句が指定されておらず、DEFINITION 句で定義されていない場合、デフォルトは、地理的空間参照系の場合は DEGREE、非地理的空間参照系の場合は NULL です。

角度測定単位は、地理的空間参照系の場合は NULL 以外、非地理的空間参照系の場合は NULL にしてください。

角度測定単位は、地理的空間参照系の場合は NULL 以外、非地理的空間参照系の場合は NULL にしてください。事前に定義された測定単位をデータベースに追加するには、sa\_install\_feature システムプロシージャを使用します。

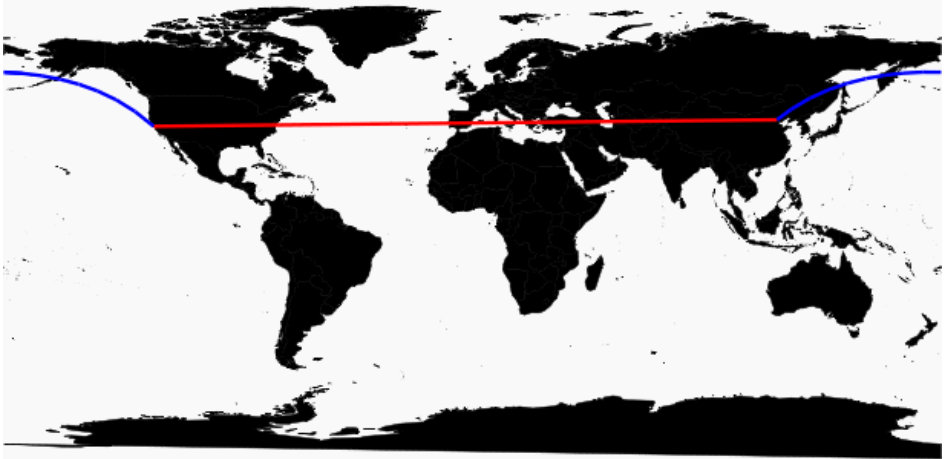
カスタム測定単位をデータベースに追加するには、CREATE SPATIAL UNIT OF MEASURE 文を使用します。

- **TYPE** – SRS で点と点を結ぶ線を解釈する方法を制御します。地理的空間参照系では、TYPE 句で ROUND EARTH (デフォルト) または PLANAR のいずれかを指定できます。ROUND EARTH モデルでは、点と点を結ぶ線が大楕円弧として解釈されます。地表面の 2 点を指定し、この 2 点および地球の中心と交差するように平面を選択したとします。この平面は地球と交差し、2 点を結ぶ線はこの交差に沿った最短距離になります。

2 点がまったく反対側にある場合、2 点および地球の中心と交差する単一でユニークな平面は存在しません。このような正反対の点を結ぶ線セグメントは無効であり、ROUND EARTH モデルでエラーになります。

ROUND EARTH モデルでは、地球が回転楕円体と見なされ、地球の湾曲に沿った線が選択されます。場合によっては、2 点間の線が x= 経度、y= 緯度の正距円筒図法における直線として解釈される平面モデルの使用が必要となる場合があります。

次の例では、青い線は ROUND EARTH モデルで使用される線の解釈を示し、赤い線は対応する PLANAR モデルを示します。



PLANAR モデルは、他の製品で使用される解釈と一致させるために使用できます。また、ROUND EARTH モデルでサポートされていないメソッド (ST\_Area、ST\_ConvexHull など) には制限があり、一部のメソッドは部分的にのみサポートされている (ST\_Distance はポイントジオメトリでのみサポートされる) ため、PLANAR モデルも便利です。円ストリングに基づいたジオメトリは、ROUND EARTH 空間参照系ではサポートされていません。

非地理的 SRS の場合には、タイプを PLANAR にしてください (これが、TYPE 句が指定されておらず、DEFINITION 句が指定されていないか非地理的定義を使用している場合のデフォルトです)。

- **COORDINATE** – 空間参照系の次元の境界。coordinate-name は、空間参照系で使用される座標系の名前です。非地理的座標系の場合は、coordinate-name に x、y、または m を使用できます。地理的座標系の場合は、coordinate-name に LATITUDE、LONGITUDE、z、または m を使用できます。

次元で境界を配置しない場合は、UNBOUNDED を指定します。境界の上限と下限を設定するには、BETWEEN 句を使用します。

X 座標と Y 座標には、関連する境界が必要です。地理的座標系の場合は、これらの設定を COORDINATE 句で上書きしないかぎり、デフォルトにより経度座標は -180 ~ 180 度の間、緯度座標は -90 ~ 90 の間になります。非地理的座標系の場合は、X 座標と Y 座標の両方について CREATE 文で境界を指定してください。

LATITUDE と LONGITUDE は、地理的座標系に使用されます。LATITUDE と LONGITUDE の境界は、指定がなければデフォルトにより地球全体になります。

- **ELLIPSOID** – タイプ ROUND EARTH の空間参照系の楕円として、地球を表現するときに使用する値。DEFINITION 句が指定されている場合は、この句で楕円定義を指定できます。ELLIPSOID 句が指定されていると、このデフォルトの楕円が上書きされます。

地球の自転が引き起こす扁平化により地球の中心から北極または南極までの距離が赤道までの距離より短くなるため、地球は完全な球体ではありません。このため、地球は、半長径 (中心から赤道までの距離) と半短径 (中心から極までの距離) に異なる値を使用する楕円としてモデル化されます。半長径と逆扁平率を使用して楕円を定義するのが最も一般的ですが、代わりに半短径を使用して指定することもできます (たとえば、完全な球形を使用して地球を近似させる場合には、この方法を使用します)。半長径と半短径は空間参照系の線形単位で定義され、逆扁平率 (1/f) は次の式で求められる比率です。

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

product-name では、地理的空間参照系で距離を計算するときに楕円定義が使用されます。

- **SNAP TO GRID** – 平らな地球 (平面) の空間参照系では、SNAP TO GRID 句を使用して、SAP Sybase IQ で計算に使用されるグリッドサイズを定義します。デフォルトでは、X と Y の空間の境界におけるすべての点で 12 有効桁数を格納できるように SAP Sybase IQ でグリッドサイズが選択されます。たとえば、空間参照系の境界が X は -180 ~ 180 の間、Y は -90 ~ 90 の間の場合、グリッドサイズ 0.000000001 (1E-9) が選択されます。

- **TOLERANCE** – 平らな地球 (平面) の空間参照系では、TOLERANCE 句を使用して、点を比較するときに使用する精度を指定します。2 点間の距離が tolerance-distance より小さい場合、その 2 点は同じと見なされます。tolerance-distance を指定することにより、入力データまたは制限された内部精度の不正確さに対する許容度を制御できます。デフォルトでは、tolerance-distance は grid-size と等しくなるように設定されます。

0 に設定すると、2 点が完全に等しい場合にのみ同じであると見なされます。

曲面の空間参照系では、TOLERANCE を 0 に設定してください。

- **POLYGON FORMAT** – SAP Sybase IQ では、まず、多角形を構成するリングの方向によって多角形が解釈されます。定義された点の順序でリングを移動した場合、多角形の内側がリングの左側になります。平面と曲面の空間参照系では、同じルールが適用されます。

SAP Sybase IQ で使用される解釈は一般的ですが、汎用的な解釈ではありません。正反対の方向を使用する製品もあれば、多角形の解釈にリングの方向を使用しない製品もあります。POLYGON FORMAT 句を使用して、必要に応じて入



力データと一致する多角形の解釈を選択できます。次の値がサポートされません。

- **CounterClockwise** – 入力 は SAP Sybase IQ の内部解釈に従います。リングの方向に従って、多角形の内側が左側になります。
- **Clockwise** – 入力 は SAP Sybase IQ の内部解釈と反対になります。リングの方向に従って、多角形の内側が右側になります。
- **EvenOdd** – (デフォルト) リングの方向は無視され、多角形の内部は代わりにリングのネストによって決定されます。外部のリングは最大のリングとなり、内部のリングはこのリングの内側の小さなリングとなります。斜線は、リング内の点から、すべてのリングを交差して外側に向かってトレースされます。交差するリングの数が偶数の場合は、外部リングです。奇数の場合は、内部リングです。
- **STORAGE FORMAT** – 空間データをデータベースにロードするときの格納内容を制御します。考えられる値は、次のとおりです。
  - **Internal** – SAP Sybase IQ は正規化された表現のみを格納します。元の入力特性を再現する必要がない場合に指定します。これは、平面 (TYPE PLANAR) の空間参照系のデフォルトです。
  - **Original** – SAP Sybase IQ は元の表現のみを格納します。元の入力特性を再現できますが、格納された値のすべての操作に対して正規化の手順を繰り返す必要があり、データの操作が遅くなる可能性があります。
  - **Mixed** – SAP Sybase IQ は内部バージョンを格納します。元のバージョンと異なる場合、SAP Sybase SQL Anywhere® は元のバージョンも格納します。両方のバージョンを格納すると、元の表現特性を再現でき、格納された値の操作に対して正規化の手順を繰り返す必要はありません。ただし、ジオメトリごとに 2 つの表現が格納される可能性があるため、記憶領域の要件が大幅に増加します。Mixed は、曲面の空間参照系 (TYPE ROUND EARTH) のデフォルトフォーマットです。

## 例

(先頭に戻る) (55 ページ)

- **例 1** – 空間参照系 mySpatialRS を作成します。

```
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+n0_defs';
```

## 使用法

(先頭に戻る) (55 ページ)

地理空間参照系では、LINEAR と ANGULAR の両方の測定単位を指定できます。非地理空間参照系では、LINEAR 測定単位のみを指定できます。LINEAR 測定単位は、点と領域間の距離を計算するために使用されます。ANGULAR 測定単位は、角度の緯度と経度を解釈する方法を示し、投影座標系の場合は NULL、地理的座標系の場合は NULL 以外になります。

操作から返される派生ジオメトリはすべて正規化されます。

SQL Anywhere 以外のデータベースで同期されるデータを操作する場合は、データの元の特性を保持できるようにするため、STORAGE FORMAT を 'Original' または 'Mixed' のいずれかに設定してください。

## 標準

(先頭に戻る) (55 ページ)

ANSI SQL – 準拠レベル： Transact-SQL® 拡張。

## パーミッション

(先頭に戻る) (55 ページ)

次のいずれかが必要です。

- MANAGE ANY SPATIAL OBJECT システム権限

- CREATE ANY OBJECT システム権限

## CREATE SPATIAL UNIT OF MEASURE 文

---

空間測定単位を作成するか、置き換えます。

クイックリンク：

「パラメータ」 (63 ページ)

「例」 (63 ページ)

「使用法」 (64 ページ)

「標準」 (64 ページ)

「パーミッション」 (64 ページ)

### 構文

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

### パラメータ

(先頭に戻る) (63 ページ)

- **OR REPLACE** – OR REPLACE の指定は、空間測定単位を作成するか、既存の空間測定単位を同じ名前で置き換えます。この句は、現在の権限を保持します。使用中の空間単位を置き換えようとすると、エラーが返されます。
- **TYPE** – 角度 (ANGULAR) または距離 (LINEAR) のどちらに測定単位を使用するかを定義します。
- **CONVERT USING** – ベース単位に関する空間単位の換算係数。線形単位の場合、ベース単位は METRE です。角度単位の場合、ベース単位は RADIANT です。

### 例

(先頭に戻る) (63 ページ)

- **例 1** – Test という空間測定単位を作成します。

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

## 使用法

(先頭に戻る) (63 ページ)

CONVERT USING 句は、定義された測定単位の値をベース測定単位(ラジアンまたはメートル)に換算する方法を定義するために使用します。指定された換算係数を測定値に乗算して、ベース測定単位の値を取得します。たとえば、測定値 512 ミリメートルに換算係数 0.001 を乗算して、測定値 0.512 メートルを取得します。

距離 (ST\_Distance または ST\_Length) または領域を計算する場合、空間参照系では常に線形測定単位が使用されます。たとえば、空間参照系の線形測定単位がマイルの場合、使用する領域単位は平方マイルになります。場合によっては、線形測定単位を使用するよう指定するオプションパラメータが空間メソッドで受け入れられることがあります。たとえば、空間参照系の線形測定単位がマイルの場合、オプションパラメータ 'metre' を使用すると、2つのジオメトリ間の距離をメートルで取得できます。

投影座標系の場合は、空間参照系の線形単位で X 座標と Y 座標が指定されます。地理的座標系の場合は、空間参照系に関連する角度測定単位で緯度と経度が指定されます。多くの場合、この角度測定単位は度ですが、有効な任意の角度測定単位を使用できます。

sa\_install\_feature システムプロシージャを使用して、事前に定義された測定単位をデータベースに追加できます。

## 標準

(先頭に戻る) (63 ページ)

ANSI SQL – 準拠レベル：Transact-SQL 拡張。

## パーミッション

(先頭に戻る) (63 ページ)

次のいずれかが必要です。

- MANAGE ANY SPATIAL OBJECT システム権限
- CREATE ANY OBJECT システム権限

## **DROP SPATIAL UNIT OF MEASURE 文**

---

空間測定単位を削除します。

クイックリンク：

「パラメータ」 (65 ページ)

「例」 (65 ページ)

「標準」 (65 ページ)

「パーミッション」 (65 ページ)

## 構文

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

## パラメータ

(先頭に戻る) (64 ページ)

- **IF EXISTS** – 存在しない空間測定単位を **DROP SPATIAL UNIT OF MEASURE** 文が削除しようとしたときにエラーを返さないようにします。

## 例

(先頭に戻る) (64 ページ)

- **例** – 次の例は、Test という名前の架空の空間測定単位を削除します。

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

## 標準

(先頭に戻る) (64 ページ)

ANSI SQL – 準拠レベル：Transact-SQL 拡張。

## パーミッション

(先頭に戻る) (64 ページ)

次のいずれかが必要です。

- **MANAGE ANY SPATIAL OBJECT** システム権限
- **DROP ANY OBJECT** システム権限
- 空間測定単位を所有していること

## **DROP SPATIAL REFERENCE SYSTEM 文**

---

空間参照系を削除します。

クイックリンク：

「パラメータ」 (66 ページ)

## 付録：SQL 文

「標準」 (66 ページ)

「パーミッション」 (66 ページ)

### 構文

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

### パラメータ

(先頭に戻る) (65 ページ)

- **IF EXISTS** – 存在しない空間参照系を **DROP SPATIAL REFERENCE SYSTEM** 文が削除しようとしたときにエラーを返さないようにします。

### 標準

(先頭に戻る) (65 ページ)

ANSI SQL – 準拠レベル：Transact-SQL 拡張。

### パーミッション

(先頭に戻る) (65 ページ)

次のいずれかが必要です。

- **MANAGE ANY SPATIAL OBJECT** システム権限
- **DROP ANY OBJECT** システム権限
- 空間参照系を所有していること

## ALTER SPATIAL REFERENCE SYSTEM 文

既存の空間参照系の設定を変更します。

クイックリンク：

「パラメータ」 (67 ページ)

「例」 (72 ページ)

「使用法」 (72 ページ)

「標準」 (72 ページ)

「パーミッション」 (72 ページ)

## 構文

### ALTER SPATIAL REFERENCE SYSTEM

```

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

srs-attribute - (構文に戻る)
    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' }

```

## パラメータ

(先頭に戻る) (66 ページ)

- **IDENTIFIED BY** – 空間参照系の SRID 番号。
- **DEFINITION** – 座標系のデフォルトを設定または上書きします。DEFINITION 句以外の句で属性が設定されている場合は、DEFINITION 句の設定内容に関係なく、他の句で指定された値が優先されます。

*definition-string* は、SQL/MM と OGC で定義されるような、空間参照系の Well Known Text 構文の文字列です。たとえば、次のクエリは WGS 84 の定義を返します。

```

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

```

Interactive SQL で戻り値をダブルクリックすると、読みやすい形で値が表示されます。

DEFINITION 句が指定されている場合は、definition-string が解析され、属性のデフォルト値の選択に使用されます。たとえば、definition-string に、organization-name と organization-srs-id を定義する AUTHORITY 要素が含まれる場合があります。

definition-string のパラメータ値は、SQL 文の句で明示的に設定した値で上書きされます。たとえば、ORGANIZATION 句が指定されていると、definition-string の ORGANIZATION の値が上書きされます。

- **ORGANIZATION** – 新しい空間参照系の基となる空間参照系を作成した組織に関する情報。
- **TRANSFORM DEFINITION** – 空間参照系に使用する変換の説明。現時点では、PROJ.4 変換のみがサポートされています。変換定義は、空間参照系間でデータを変換するときに ST\_Transform メソッドで使用されます。一部の変換は、transform-definition-string が定義されていなくても使用できます。
- **LINEAR UNIT OF MEASURE** – 空間参照系の線形測定単位。指定する値は、ST\_UNITS\_OF\_MEASURE システムビューで定義された線形測定単位と一致している必要があります。

この句が指定されておらず、DEFINITION 句で定義されていない場合、デフォルトは METRE です。事前に定義された測定単位をデータベースに追加するには、sa\_install\_feature システムプロシージャを使用します。

カスタム測定単位をデータベースに追加するには、CREATE SPATIAL UNIT OF MEASURE 文を使用します。

---

**注意：** METRE と METER はいずれも有効なスペルですが、SQL/MM 標準に準拠している METRE を使用することをおすすめします。

---

- **ANGULAR UNIT OF MEASURE** – 空間参照系の角度測定単位。指定する値は、ST\_UNITS\_OF\_MEASURE システムテーブルで定義された角度測定単位と一致している必要があります。

この句が指定されておらず、DEFINITION 句で定義されていない場合、デフォルトは、地理的空間参照系の場合は DEGREE、非地理的空間参照系の場合は NULL です。

角度測定単位は、地理的空間参照系の場合は NULL 以外、非地理的空間参照系の場合は NULL にしてください。



角度測定単位は、地理的空間参照系の場合は NULL 以外、非地理的空間参照系の場合は NULL にしてください。事前に定義された測定単位をデータベースに追加するには、sa\_install\_feature システムプロシージャを使用します。

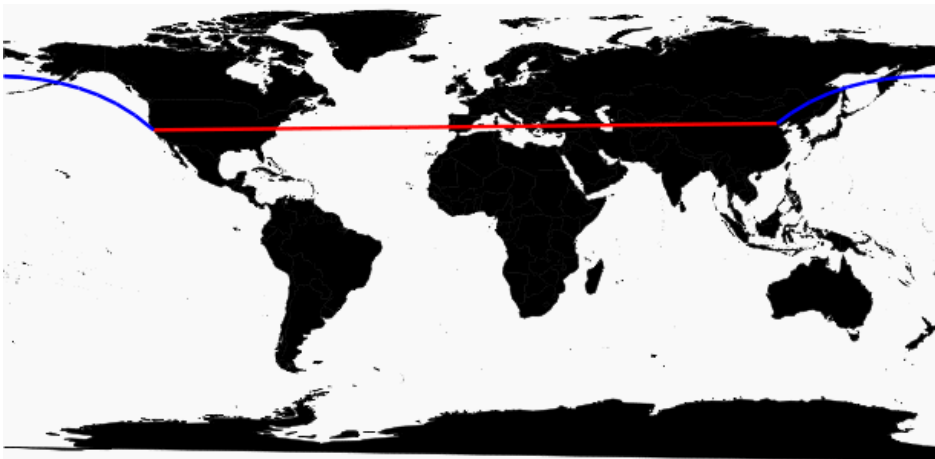
カスタム測定単位をデータベースに追加するには、CREATE SPATIAL UNIT OF MEASURE 文を使用します。

- **TYPE – SRS** で点と点を結ぶ線を解釈する方法を制御します。地理的空間参照系では、TYPE 句で ROUND EARTH (デフォルト) または PLANAR のいずれかを指定できます。ROUND EARTH モデルでは、点と点を結ぶ線が大楕円弧として解釈されます。地表面の 2 点を指定し、この 2 点および地球の中心と交差するように平面を選択したとします。この平面は地球と交差し、2 点を結ぶ線はこの交差に沿った最短距離になります。

2 点がまったく反対側にある場合、2 点および地球の中心と交差する単一でユニークな平面は存在しません。このような正反対の点を結ぶ線セグメントは無効であり、ROUND EARTH モデルでエラーになります。

ROUND EARTH モデルでは、地球が回転楕円体と見なされ、地球の湾曲に沿った線が選択されます。場合によっては、2 点間の線が x= 経度、y= 緯度の正距円筒図法における直線として解釈される平面モデルの使用が必要となる場合があります。

次の例では、青い線は ROUND EARTH モデルで使用される線の解釈を示し、赤い線は対応する PLANAR モデルを示します。



PLANAR モデルは、他の製品で使用される解釈と一致させるために使用できます。また、ROUND EARTH モデルでサポートされていないメソッド (ST\_Area、ST\_ConvexHull など) には制限があり、一部のメソッドは部分的にのみサポートされている (ST\_Distance はポイントジオメトリでのみサポートされる) ため、

PLANAR モデルも便利です。円ストリングに基づいたジオメトリは、ROUND EARTH 空間参照系ではサポートされていません。

非地理的 SRS の場合には、タイプを PLANAR にしてください (これが、TYPE 句が指定されておらず、DEFINITION 句が指定されていないか非地理的定義を使用している場合のデフォルトです)。

- **COORDINATE** – 空間参照系の次元の境界。coordinate-name は、空間参照系で使用される座標系の名前です。非地理的座標系の場合は、coordinate-name に x、y、または m を使用できます。地理的座標系の場合は、coordinate-name に LATITUDE、LONGITUDE、z、または m を使用できます。

次元で境界を配置しない場合は、UNBOUNDED を指定します。境界の上限と下限を設定するには、BETWEEN 句を使用します。

X 座標と Y 座標には、関連する境界が必要です。地理的座標系の場合は、これらの設定を COORDINATE 句で上書きしないかぎり、デフォルトにより経度座標は -180 ~ 180 度の間、緯度座標は -90 ~ 90 の間になります。非地理的座標系の場合は、X 座標と Y 座標の両方について CREATE 文で境界を指定してください。

LATITUDE と LONGITUDE は、地理的座標系に使用されます。LATITUDE と LONGITUDE の境界は、指定がなければデフォルトにより地球全体になります。

- **ELLIPSOID** – タイプ ROUND EARTH の空間参照系の楕円として、地球を表現するときに使用する値。DEFINITION 句が指定されている場合は、この句で楕円定義を指定できます。ELLIPSOID 句が指定されていると、このデフォルトの楕円が上書きされます。

地球の自転が引き起こす扁平化により地球の中心から北極または南極までの距離が赤道までの距離より短くなるため、地球は完全な球体ではありません。このため、地球は、半長径 (中心から赤道までの距離) と半短径 (中心から極までの距離) に異なる値を使用する楕円としてモデル化されます。半長径と逆扁平率を使用して楕円を定義するのが最も一般的ですが、代わりに半短径を使用して指定することもできます (たとえば、完全な球形を使用して地球を近似させる場合には、この方法を使用します)。半長径と半短径は空間参照系の線形単位で定義され、逆扁平率 (1/f) は次の式で求められる比率です。

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

product-name では、地理的空間参照系で距離を計算するときに楕円定義が使用されます。

- **SNAP TO GRID** – 平らな地球 (平面) の空間参照系では、SNAP TO GRID 句を使用して、SAP Sybase IQ で計算に使用されるグリッドサイズを定義します。デフォルトでは、X と Y の空間の境界におけるすべての点で 12 有効桁数を格納

できるように SAP Sybase IQ でグリッドサイズが選択されます。たとえば、空間参照系の境界が X は -180 ~ 180 の間、Y は -90 ~ 90 の間の場合、グリッドサイズ 0.000000001 (1E-9) が選択されます。

- **TOLERANCE** – 平らな地球 (平面) の空間参照系では、TOLERANCE 句を使用して、点を比較するとき使用する精度を指定します。2 点間の距離が tolerance-distance より小さい場合、その 2 点は同じと見なされます。tolerance-distance を指定することにより、入力データまたは制限された内部精度の不正確さに対する許容度を制御できます。デフォルトでは、tolerance-distance は grid-size と等しくなるように設定されます。

0 に設定すると、2 点が完全に等しい場合にのみ同じであると見なされます。

曲面の空間参照系では、TOLERANCE を 0 に設定してください。

- **POLYGON FORMAT** – SAP Sybase IQ では、まず、多角形を構成するリングの方向によって多角形が解釈されます。定義された点の順序でリングを移動した場合、多角形の内側がリングの左側になります。平面と曲面の空間参照系では、同じルールが適用されます。

SAP Sybase IQ で使用される解釈は一般的ですが、汎用的な解釈ではありません。正反対の方向を使用する製品もあれば、多角形の解釈にリングの方向を使用しない製品もあります。POLYGON FORMAT 句を使用して、必要に応じて入力データと一致する多角形の解釈を選択できます。次の値がサポートされます。

- **CounterClockwise** – 入力 は SAP Sybase IQ の内部解釈に従います。リングの方向に従って、多角形の内側が左側になります。
- **Clockwise** – 入力 は SAP Sybase IQ の内部解釈と反対になります。リングの方向に従って、多角形の内側が右側になります。
- **EvenOdd** – (デフォルト) リングの方向は無視され、多角形の内部は代わりにリングのネストによって決定されます。外部のリングは最大のリングとなり、内部のリングはこのリングの内側の小さなリングとなります。斜線は、リング内の点から、すべてのリングを交差して外側に向かってトレースされます。交差するリングの数が偶数の場合は、外部リングです。奇数の場合は、内部リングです。
- **STORAGE FORMAT** – 空間データをデータベースにロードするときの格納内容を制御します。考えられる値は、次のとおりです。
  - **Internal** – SAP Sybase IQ は正規化された表現のみを格納します。元の入力特性を再現する必要がない場合に指定します。これは、平面 (TYPE PLANAR) の空間参照系のデフォルトです。

- **Original** – SAP Sybase IQ は元の表現のみを格納します。元の入力特性を再現できますが、格納された値のすべての操作に対して正規化の手順を繰り返す必要があります、データの操作が遅くなる可能性があります。
- **Mixed** – SAP Sybase IQ は内部バージョンを格納します。元のバージョンと異なる場合、SAP Sybase SQL Anywhere® は元のバージョンも格納します。両方のバージョンを格納すると、元の表現特性を再現でき、格納された値の操作に対して正規化の手順を繰り返す必要はありません。ただし、ジオメトリごとに 2 つの表現が格納される可能性があるため、記憶領域の要件が大幅に増加します。Mixed は、曲面の空間参照系 (TYPE ROUND EARTH) のデフォルトフォーマットです。

## 例

(先頭に戻る) (66 ページ)

- **例** – mySpatialRef という架空の空間参照系の多角形フォーマットを EvenOdd に変更します。

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

## 使用法

(先頭に戻る) (66 ページ)

空間参照系を参照している既存のデータがある場合、その空間参照系を変更することはできません。たとえば、ST\_Point(SRID=8743) として宣言したカラムがある場合、SRID 8743 を持つ空間参照系は変更できません。これは、格納フォーマットなどの空間参照系の多くの属性が、データの格納フォーマットに影響を及ぼすためです。SRID を参照するデータがある場合は、新しい空間参照系を作成して、データを新しい SRID に変換します。

## 標準

(先頭に戻る) (66 ページ)

ANSI SQL – 準拠レベル： Transact-SQL 拡張。

## パーミッション

(先頭に戻る) (66 ページ)

次のいずれかが必要です。

- その空間参照系を所有している
- 空間参照系に対する ALTER 権限

- `MANAGE ANY SPATIAL OBJECT` システム権限
- `ALTER ANY OBJECT` システム権限

## ALTER TABLE 文

---

テーブルの定義を変更します。

クイックリンク：

「パラメータ」 (75 ページ)

「例」 (84 ページ)

「使用法」 (87 ページ)

「標準」 (87 ページ)

「パーミッション」 (87 ページ)

### 構文

構文 1 - Alter Owner

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

構文 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-schemecomposite-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

```

## パラメータ

(先頭に戻る) (73 ページ)

- { **ENABLE** | **DISABLE** } **RLV STORE** - リアルタイムのインメモリ更新用に、このテーブルを RLV ストアに登録します。IQ テンポラリテーブルまたはマルチ

プレックス環境ではサポートされていません。この値は、データベースオプション **BASE\_TABLES\_IN\_RLV** の値よりも優先されます。

- **ADD column-definition [ column-constraint ]** – テーブルに新しいカラムを追加します。

NOT NULL を指定するには、テーブルが空であることが必要です。IDENTITY カラムまたは DEFAULT AUTOINCREMENT カラムの追加時に、テーブルにデータが含まれていてもかまいません。カラムにデフォルトの IDENTITY 値が指定されていれば、新しいカラムのすべてのローに連続する値が入力されます。また、1つのカラムキーに、FOREIGN 制約をカラム制約として追加できます。IDENTITY/DEFAULT AUTOINCREMENT カラムの値は、テーブル内の各ローをユニークに識別します。

IDENTITY/DEFAULT AUTOINCREMENT カラムには、挿入や更新の際に自動的に生成される連続した数値が格納されます。DEFAULT AUTOINCREMENT カラムは、IDENTITY カラムとも呼ばれます。IDENTITY/DEFAULT AUTOINCREMENT を使用するカラムは、整数データ型のいずれか、または位取りが0の真数値型であることが必要です。カラム制約と IDENTITY/DEFAULT AUTOINCREMENT カラムの詳細については、「CREATE TABLE 文」を参照してください。

**IQ UNIQUE** 制約 - カラムの予期されるカーディナリティを定義し、カラムをフラット FP または NBit FP のどちらとしてロードするかを決定します。IQ UNIQUE(*n*) の値を明示的に 0 に設定すると、カラムはフラット FP としてロードされます。IQ UNIQUE 制約のないカラムは、FP\_NBIT\_AUTOSIZE\_LIMIT、FP\_NBIT\_LOOKUP\_MB、および FP\_NBIT\_ROLLOVER\_MAX\_MB オプションで定義された上限まで暗黙的に NBit としてロードされます。

IQ UNIQUE を FP\_NBIT\_AUTOSIZE\_LIMIT 未満の *n* 値とともに使用する必要はありません。自動サイズ機能によって、カーディナリティが低いか中程度のカラムはすべて NBit としてサイズ決定されます。カラムをフラット FP としてロードする場合や、重複しない値の数が FP\_NBIT\_AUTOSIZE\_LIMIT を超えるときにカラムを NBit としてロードする場合は、IQ UNIQUE を使用します。

---

**注意：**

- 高い IQ UNIQUE 値を指定する際はメモリ使用率を考慮します。マシンリソースに制限がある場合、FP\_NBIT\_ENFORCE\_LIMITS='OFF' (デフォルト) でロードしないでください。  
SAP Sybase IQ 16.0 以前は、16777216 を超える IQ UNIQUE *n* 値はフラット FP にロールオーバーされていました。16.0 では、より大きい IQ UNIQUE 値が



トークン化でサポートされていますが、カーディナリティとカラム幅に応じて、重大なメモリリソース要件が発生する場合があります。

- BIT、BLOB、および CLOB の各データ型は NBit ディクショナリ圧縮をサポートしません。FP\_NBIT\_IQ15\_COMPATIBILITY='OFF' である場合、これらのデータ型を含む CREATE TABLE 文または ALTER TABLE 文でゼロ以外の IQ UNIQUE カラムを指定すると、エラーが返されます。

- **ALTER *column-name* *column-alteration*** – カラム定義を次のように変更します。
  - **SET DEFAULT *default-value*** - テーブルの既存のカラムのデフォルト値を変更します。この作業では MODIFY 句も使用できますが、ALTER は ISO/ANSI SQL に準拠しているのに対して MODIFY は準拠していません。デフォルト値を変更しても、テーブルの既存の値は変更されません。
  - **DROP DEFAULT** - テーブルの既存のカラムのデフォルト値を削除します。この作業では MODIFY 句も使用できますが、ALTER は ISO/ANSI SQL に準拠しているのに対して MODIFY は準拠していません。デフォルトを削除しても、テーブルの既存の値は変更されません。
  - **ADD** - 名前付き制約または CHECK 条件をカラムに追加します。新しい制約または条件は、それを定義した後でテーブルに対して実行される処理のみに適用されます。テーブルの既存の値は、新しい制約や条件を満たすかどうかの検証を受けません。
  - **CONSTRAINT *column-constraint-name*** - オプションのカラム制約名を指定すると、後で、カラム制約全体を修正するのではなく、制約を個別に修正または削除できます。
  - **[ CONSTRAINT *constraint-name* ] CHECK ( *condition* )** - この句はカラムに検査制約を追加するときを使用します。
  - **SET COMPUTE ( *expression* )** - 計算カラムに関連付けられた式を変更します。この文を実行すると、カラムの値が再計算されます。新しい式が無効な場合、この文は失敗します。
  - **DROP COMPUTE** - 計算カラムから非計算カラムに変更します。この文はテーブル内の既存の値を変更しません。
- **ADD *table-constraint*** – テーブルに制約を追加します。

シングルカラムまたはマルチカラムのキーに、外部キー制約をテーブル制約として追加することもできます。PRIMARY KEY を指定する場合、テーブルには CREATE TABLE 文または別の ALTER TABLE 文で作成したプライマリキーがあってはなりません。テーブル制約の詳細については、「CREATE TABLE 文」を参照してください。

**注意：** テーブルまたはカラムの制約は MODIFY (変更) できません。制約を変更するには、古い制約を DELETE (削除) し、新しい制約を ADD (追加) します。

- **DROP *drop-object*** - テーブルオブジェクトを削除します。
  - **DROP *column-name*** - カラムをテーブルから削除します。カラムがマルチカラムインデックス、一意性制約、外部キー、またはプライマリキーに含まれている場合は、インデックス、制約またはキーを削除してからカラムを削除してください。このようにするとカラムを参照する検査制約は削除されません。IDENTITY/DEFAULT AUTOINCREMENT カラムを削除できるのは、テーブルの IDENTITY\_INSERT が OFF に設定され、かつテーブルがローカルテンポラリテーブルでない場合だけです。
  - **DROP CHECK** - テーブルのすべての検査制約を削除します。テーブル検査制約とカラム検査制約の両方が対象となります。
  - **DROP CONSTRAINT *constraint-name*** - テーブルまたは指定したカラムの名前付き制約を削除します。
  - **DROP UNIQUE ( *column-name, ...* )** - 指定したカラムの一意性制約を削除します。一意性制約を参照する外部キー (プライマリキーではなく) も削除します。関連する外部キー制約がある場合は、エラーが報告されます。ALTER TABLE を使用して、プライマリキーを参照するすべての外部キーを削除した後でなければ、プライマリキー制約を削除することはできません。
  - **DROP PRIMARY KEY** - プライマリキーを削除します。このテーブルのプライマリキーを参照するすべての外部キーも削除します。関連する外部キー制約がある場合は、エラーが報告されます。プライマリキーに強制力がない場合、そのプライマリキーに強制力のない外部キー制約が存在すると、DELETE はエラーを返します。
  - **DROP FOREIGN KEY *role-name*** - 特定のロール名を持つ、該当テーブルの外部キー制約を削除します。その外部キー制約に対して自動的に作成された、ユニークでない HG インデックスは削除されません。HG インデックスは、DROP INDEX 文を使用して明示的に削除してください。
  - **DROP [ PARTITION ]** - 指定したパーティションを削除します。パーティション P1 内のローとパーティション定義が削除されます。最後のパーティションは削除できません。これは、分割されたテーブルが非分割テーブルに変換されるためです (分割されたテーブルをマージするには、UNPARTITION 句を代わりに使用します)。次に例を示します。

```
CREATE TABLE foo (c1 INT, c2 INT)
PARTITION BY RANGE (c1)
(P1 VALUES <= (100) IN dbosp1,
 P2 VALUES <= (200) IN dbosp2,
 P3 VALUES <= (MAX) IN dbosp3
) IN dbosp4);
LOAD TABLE ...
ALTER TABLE DROP PARTITION P1;
```

- **RENAME *rename-object*** - テーブル内のオブジェクトの名前を変更します。

- **RENAME** *new-table-name* - テーブルの名前を *new-table-name* に変更します。古いテーブル名を使用しているアプリケーションは、修正する必要があります。また、古いテーブル名と同じ名前が自動的に割り当てられた外部キーの名前は、変更しません。
- **RENAME** *column-name* **TO** *new-column-name* - カラムの名前を *new-column-name* に変更します。古いカラム名を使用しているアプリケーションがある場合は、修正が必要になります。
- **RENAME** [ **PARTITION** ] - 既存のパーティション名を変更します。
- **RENAME** *constraint-name* **TO** *new-constraint-name* - 制約の名前を *new-constraint-name* に変更します。古い制約名を使用しているアプリケーションがある場合は、修正が必要になります。
- **MOVE clause** - テーブルオブジェクトを移動します。各テーブルオブジェクトは 1 つの DB 領域にのみ置くことができます。どのタイプの ALTER MOVE も、移動中はテーブルへの変更をすべてブロックします。

---

**注意：** オブジェクトをキャッシュ DB 領域に移動することはできません。

---

- **MOVE TO** - テーブルが新しい DB 領域にマッピングされると、そのテーブルと同じ DB 領域に存在するカラム、インデックス、一意性制約、プライマリキー、外部キー、メタデータなどのすべてのテーブルオブジェクトを新しい DB 領域に移動します。ALTER Column MOVE TO 句は、分割テーブルに対しては要求できません。  
BIT データ型のカラムは DB 領域に明示的に配置することはできません。以下は BIT データ型に対してサポートされていません。  

```
ALTER TABLE t2 alter c1_bit MOVE TO iq_main;
```
- **MOVE TABLE METADATA** - テーブルのメタデータを新しい DB 領域に移動します。分割されたテーブルでは、MOVE TABLE METADATA はパーティション間で共有されるメタデータも移動します。
- **MOVE PARTITION** - 指定したパーティションを新しい DB 領域に移動します。
- **PARTITION BY** - 大きなテーブルを、より小さく管理しやすいストレージオブジェクトに分割します。各パーティションは親テーブルと同じ論理属性を共有しますが、別々の DB 領域に配置して個別に管理できます。SAP Sybase IQ は、次のような複数のテーブル分割スキームをサポートしています。
  - ハッシュパーティション
  - 範囲パーティション
  - 複合パーティション

partition-key は、テーブル分割キーが格納されている 1 つまたは複数のカラムです。分割キーには、NULL 値 および DEFAULT 値を含めることができますが、次のカラムを含めることはできません。

- LOB (BLOB または CLOB) カラム
- BINARY または VARBINARY カラム
- 長さが 255 バイトを超える CHAR または VARCHAR カラム
- BIT カラム
- FLOAT/DOUBLE/REAL カラム
- **PARTITION BY RANGE** – 分割カラム内の値の範囲によってローを分割します。範囲分割は、単一の分割キーカラムおよび最大 1024 パーティションまでに制限されています。range-partitioning-scheme 内の partition-key は、テーブル分割キーが格納されているカラムです。

```
range-partition-decl:
  partition-name VALUES <= ( {constant-expr | MAX } [ ,
  { constant-expr | MAX } ]... )
  [ IN dbspace-name ]
```

partition-name は、テーブルローが格納される新しいパーティションの名前です。パーティション名は、テーブル上にあるパーティションセット内でユニークである必要があります。パーティション名は必須です。

- **VALUE** – 各パーティションの包括的な上限を (昇順に) 指定します。ユーザは、各ローが 1 つのパーティションのみに分配されるように、各範囲分割の分割基準を指定する必要があります。NULL は分割カラムに使用でき、NULL を分割キー値に含んだローは最初のテーブル分割に属します。ただし、NULL をバインド値に指定することはできません。

最初のパーティションには、下限 (MIN 値) は設定されていません。分割キーの最初のカラムにある NULL セルのローは、最初のパーティションに移動します。最後のパーティションでは、包括的な上限または MAX を指定できます。最後のパーティションの上限値が MAX でない場合は、最後のパーティションの上限値よりも大きい分割キーの値を含んだローをロードまたは挿入すると、エラーが生成されます。

- **MAX** – 無制限の上限を示し、最後のパーティションに対してのみ指定できます。
- **IN** – partition-decl でパーティションのローが存在する DB 領域を指定します。

次の制限を設定すると、範囲分割されたテーブルの分割キーとバインド値がその影響を受けます。

- 未分割のテーブルを範囲分割できるのは、すべての既存のローが最初のパーティションに属する場合のみです。
- パーティションバインドは定数式でなく、定数として指定する必要があります。

- パーティションバインドは、パーティションの作成順に応じて、昇順で指定する必要があります。つまり、2 番目のパーティションの上限は最初のパーティションよりも高く指定する必要があります、というようになります。さらに、パーティションバインドの値は、対応する分割キーカラムのデータ型と互換性がなければなりません。たとえば、VARCHAR は CHAR と互換性があります。
- バインド値に対応する分割キーのカラムとは異なるデータ型が指定されていると、SAP Sybase IQ はバインド値を分割キーのカラムのデータ型に変換します。ただし、次の場合は例外となります。
- 明示的な変換は使用できません。この例では、INT から VARCHAR に明示的に変換しようとしてエラーが生成されます。

```
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)))
```

- データロスにつながる暗黙的な変換は使用できません。この例では、パーティションバインドは分割キー型と互換性がありません。丸めを前提で処理を行うとデータロスにつながる可能性があり、エラーが生成されます。
- この例では、パーティションバインドと分割キーのデータ型の間には互換性があります。バインド値は FLOAT 値に直接変換されます。丸め処理は必要なく、変換はサポートされています。

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

- 非バイナリデータ型からバイナリデータ型に変換することはできません。たとえば、次の変換は実行できずに、エラーが返されます。

```
CREATE TABLE newemp (name BINARY)
PARTITION BY RANGE(name)
(p1 VALUES <= ("Maarten"),
p2 VALUES <= ("Zymerman"))
```

- NULL を範囲分割テーブルで境界として使用することはできません。
- 分割キーの最初のカラムのセル値が NULL と評価された場合、ローは最初のパーティションに挿入されます。SAP Sybase IQ は、1 つのカラムの分割キーのみをサポートしているため、分割キー内に NULL が含まれていると、ローは最初のパーティションに分配されます。
- PARTITION BY HASH** – 内部ハッシュ関数によって処理された分割キーの値に基づいて、データをパーティションにマップします。ハッシュ分割キーは最大 8 カラムで、組み合わせた宣言カラム幅が 5300 バイト以下に制限されています。ハッシュパーティションの場合、テーブル作成者は分割キーカラムのみを決定します。パーティションの数と位置は内部的に決定されます。

hash-partitioning 宣言内の partition-key は、1つのカラムまたはカラムのグループです。その複合値によってデータの各ローが格納されるパーティションが決まります。

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

• **制限事項** –

- ハッシュ分割できるのはベーステーブルのみです。グローバルテンポラリテーブルやローカルテンポラリテーブルを分割しようとすると、エラーが発生します。
  - 未分割のテーブルをハッシュ分割できるのは、そのテーブルが空の場合のみです。
  - ハッシュパーティションの追加、削除、マージ、分割はできません。
  - カラムをハッシュ分割キーから追加または削除することはできません。
- **PARTITION BY HASH RANGE** – ハッシュ分割されたテーブルを範囲によってさらに分割します。hash-range-partitioning-scheme 宣言内の SUBPARTITION BY RANGE 句は、新しい範囲サブパーティションを既存のハッシュ範囲分割テーブルに追加します。

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

ハッシュパーティションはデータの論理的な配分および配置方法を指定するのに対して、範囲サブパーティションはデータの物理的な配置方法を指定します。新しい範囲サブパーティションは、既存のハッシュ範囲分割テーブルと同じハッシュ分割キーを持つハッシュによって論理的に分割されます。範囲サブパーティションは1つのカラムに制限されています。

• **制限事項** –

- ハッシュ分割できるのはベーステーブルのみです。グローバルテンポラリテーブルやローカルテンポラリテーブルを分割しようとすると、エラーが発生します。
- ハッシュ分割されたテーブルを範囲によってさらに分割できるのは、そのテーブルが空の場合のみです。
- ハッシュパーティションの追加、削除、マージ、分割はできません。
- カラムをハッシュ分割キーから追加または削除することはできません。

---

**注意：** 範囲パーティションと複合分割スキームは、ハッシュ範囲パーティションと同様、個別にライセンスが必要な VLDB Management オプションを必要とします。

---

- **MERGE PARTITION** – *partition-name-1* を *partition-name-2* にマージします。2つのパーティションが隣接しており、データが同じ DB 領域にある場合は、パー

パーティションをマージできます。低いパーティションの値を持つパーティションを高いパーティションの値を持つ隣接パーティションにマージする場合のみ、パーティションをマージできます。サーバでは、パーティションのマージ先の DB 領域に対する CREATE 権限がチェックされないことに注意してください。隣接するパーティションを作成する方法の例については、「CREATE TABLE 文」の例を参照してください。

- **RENAME PARTITION** – 既存の PARTITION の名前を変更します。
- **UNPARTITION** – 分割されたテーブルからパーティションを削除します。各カラムは 1 つの DB 領域に配置されます。サーバでは、すべてのパーティションのデータの移動先となる DB 領域に対する CREATE 権限がチェックされないことに注意してください。ALTER TABLE UNPARTITION は、データベースのアクティビティをすべてブロックします。
- **ALTER OWNER** – テーブルの所有者を変更します。ALTER OWNER 句を ALTER TABLE 文の他の [alter-clause] 句と組み合わせて使用することはできません。
  - **[ PRESERVE | DROP ] PERMISSIONS** — 新しい所有者に古い所有者と同じ権限を付与しない場合は、DROP privileges 句 (デフォルト) を指定して、明示的に付与されていたテーブルへのユーザアクセスを許可する権限をすべて取り消すことができます。そのテーブルの所有者の権限が暗黙的に与えられたものである場合、その権限が新しい所有者に付与され、前の所有者から削除されます。
  - **[ PRESERVE | DROP ] FOREIGN KEYS** - 新しい所有者が参照先テーブル内のデータにアクセスできないようにするには、DROP FOREIGN KEYS 句 (デフォルト) を使用して、テーブル内のすべての外部キーと、そのテーブルを参照しているすべての外部キーを削除します。新しい所有者がすべての参照元テーブルを所有していないかぎり、PRESERVE FOREIGN KEYS 句と DROP PERMISSIONS 句を組み合わせると失敗します。

ALTER TABLE ALTER OWNER 文は次の場合に失敗します。

- 元のテーブルと同じ名前を持つ別のテーブルが存在し、新しいユーザがそのテーブルを所有している場合。
- PRESERVE FOREIGN KEYS 句と PRESERVE PERMISSIONS 句の両方が指定され、新しいテーブル所有者以外のユーザが所有する外部キーが存在し、その外部キーが暗黙的に付与された権限 (テーブルの所有者に与えられた権限など) に依存するテーブルを参照している場合。この失敗を回避するには、参照元テーブルの元の所有者に対して SELECT 権限を明示的に付与するか、該当する外部キーを削除します。
- PRESERVE FOREIGN KEYS が指定され、PRESERVE PERMISSIONS が指定されていないときに、新しいテーブル所有者以外のユーザが所有する外部

キーが存在し、その外部キーがテーブルを参照している場合。この失敗を回避するには、該当する外部キーを削除します。

- **PRESERVE FOREIGN KEYS** 句が指定され、暗黙的に付与された権限 (テーブルの所有者に与えられた権限など) に依存する外部キーがテーブルに含まれている場合。この失敗を回避するには、新しい所有者に対して参照先テーブルの **SELECT** 権限を明示的に付与するか、該当する外部キーを削除します。
- テーブル内にシーケンスを参照するデフォルト値を持つカラムがあり、シーケンスジェネレータの **USAGE** 権限が暗黙的に付与された権限 (シーケンスの所有者に与えられた権限など) に依存する場合。この失敗を回避するには、テーブルの新しい所有者に対してシーケンスジェネレータの **USAGE** 権限を明示的に付与します。
- 元のテーブルに依存する有効化されたマテリアライズドビューが存在する場合。

## 例

(先頭に戻る) (73 ページ)

- **例 1** – 従業員の勤務先の事務所を示す新しいカラムを、Employees テーブルに追加します。

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

- **例 2** – office カラムを Employees テーブルから削除します。

```
ALTER TABLE Employees
DROP office
```

- **例 3** – Customers テーブルにカラムを追加して、各顧客の販売担当を割り当てます。

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

- **例 4** – 新しいカラム CustomerNum を Customers テーブルに追加して、デフォルト値の 88 を割り当てます。

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

- **例 5** – c2、c4、および c5 の **FP** インデックスを DB 領域 Dsp3 から Dsp6 に移動します。c1 の **FP** インデックスは Dsp1 に残ります。c3 の **FP** インデックスは Dsp2 に残ります。c5 のプライマリキーは Dsp4 に残ります。**DATE** インデックス c4\_date は Dsp5 に残ります。

```
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;

```

- **例 6 – FP** インデックス `c1` のみを DB 領域 `Dsp1` から `Dsp7` に移動します。

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

- **例 7 – 多数の ALTER TABLE** 句を使用して、パーティションの移動、分割、名前変更、およびマージを行います。

分割されたテーブルを作成します。

```

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');

```

パーティション `p2` を `dbbsp5` に移動します。

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

パーティション `p4` を 2 つのパーティションに分割します。

```

ALTER TABLE bar SPLIT PARTITION p4 INTO
(p41 VALUES <= ('2008-06-30') IN dbbsp4,
 p42 VALUES <= ('2008-12-31') IN dbbsp4);

```

次の **SPLIT PARTITION** では、データを移動する必要があるため、エラーが報告されます。既存のローが分割後にすべて同じパーティションにあるとはかぎりません。

```

ALTER TABLE bar SPLIT PARTITION p3 INTO
(p31 VALUES <= ('2007-06-30') IN dbbsp3,
 p32 VALUES <= ('2007-12-31') IN dbbsp3);

```

次のエラーが報告されます。

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

次の **SPLIT PARTITION** では、パーティションの境界値が変更されるため、エラーが報告されます。

```
ALTER TABLE bar SPLIT PARTITION p2 INTO
  (p21 VALUES <= ('2006-06-30') IN dbbsp2,
   P22 VALUES <= ('2006-12-01') IN dbbsp2);
```

次のエラーが報告されます。

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

パーティション p3 を p2 にマージします。高い境界値から低い境界値のパーティションへのマージは使用できないため、エラーが報告されます。

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

次のエラーが報告されます。

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

パーティション p2 を p3 にマージします。

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

パーティション p1 の名前を p1\_new に変更します。

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

テーブル bar の分割を解除します。

```
ALTER TABLE bar UNPARTITION;
```

テーブル bar を分割します。このコマンドでは、すべてのローが最初のパーティションに含まれている必要があるため、エラーが報告されます。

```
ALTER TABLE bar PARTITION BY RANGE (c2)
  (p1 VALUES <= ('2005-12-31') IN dbbsp1,
   P2 VALUES <= ('2006-12-31') IN DBSP2,
   P3 VALUES <= ('2007-12-31') IN dbbsp3,
   P4 VALUES <= ('2008-12-31') IN dbbsp4);
```

次のエラーが報告されます。

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

テーブル bar を分割します。

```
ALTER TABLE bar PARTITION BY RANGE (c2)
  (p1 VALUES <= ('2008-12-31') IN dbbsp1,
   P2 VALUES <= ('2009-12-31') IN dbbsp2,
   P3 VALUES <= ('2010-12-31') IN dbbsp3,
   P4 VALUES <= ('2011-12-31') IN dbbsp4);
```

- **例 8** – テーブル tab1 を変更して、インメモリリアルタイム更新の対象として RLV ストアに登録されないようにします。

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

## 使用法

(先頭に戻る) (73 ページ)

ALTER TABLE 文は、以前作成したテーブルのテーブル属性(カラム定義、制約)を変更します。構文では複数の ALTER 句を使用できますが、1つの ALTER TABLE 文の中では1つの table-constraint または column-constraint しか追加、修正、削除できないことに注意してください。ALTER TABLE 文は、他の接続で現在使用中のテーブルに影響を及ぼす場合は実行できません。ALTER TABLE は処理に時間がかかり、この文の処理中は、同じテーブルを参照する要求がサーバで処理されません。

---

**注意：** ローカルテナポラリテーブルは変更できませんが、グローバルテナポラリテーブルは、テーブルを使用する接続が1つだけの場合には変更できます。

---

SAP Sybase IQ は、REFERENCES 制約と CHECK 制約を適用します。ALTER TABLE 文で追加されるテーブルまたはカラムの検査制約は、追加される新しいカラムのいずれかに対して定義された場合にのみ、ALTER TABLE 操作の中で評価されます。CHECK 制約の詳細については、「CREATE TABLE 文」を参照してください。

ビュー定義に **SELECT \*** を使用し、その **SELECT \*** で参照されるテーブルを変更する場合は、**ALTER VIEW <viewname> RECOMPILE** を実行してビュー定義を訂正し、ビューをクエリしたときに予期しない結果が返されるのを防ぐ必要があります。

### 関連する動作:

- オートコミット。ALTER と DROP オプションは現在の接続に対するすべてのカーソルをクローズします。Interactive SQL のデータウィンドウもクリアされます。
- ALTER TABLE 操作の開示時にチェックポイントを実行します。
- カラムまたはテーブルを変更すると、その変更したカラムを参照するストアドプロシージャ、ビューなどは機能しなくなります。

## 標準

(先頭に戻る) (73 ページ)

- SQL - ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase Database 製品 - 一部の句は SAP Adaptive Server® Enterprise でサポートされています。

## パーミッション

(先頭に戻る) (73 ページ)

### 構文 1

次のいずれかが必要です。

- ALTER ANY TABLE システム権限
- ALTER ANY OBJECT システム権限
- そのテーブルに対する ALTER 権限
- テーブルを所有していること

## 構文 2

構文 1 に必要なシステム権限は、使用する句によって異なります。

句	必要な権限
ADD	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• 基礎となるテーブルに対する ALTER 権限</li> <li>• 基本となるテーブルを所有していること</li> </ul> <p>UNIQUE、PRIMARY KEY、FOREIGN KEY、または IQ UNIQUE カラム制約には、上記とともに基本となるテーブルの REFERENCES 権限が必要。</p> <p>FOREIGN KEY テーブル制約には、上記とともに次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• CREATE ANY INDEX システム権限</li> <li>• CREATE ANY OBJECT システム権限</li> <li>• ベーステーブルに対する REFERENCES 権限</li> </ul> <p>PARTITION BY RANGE には、上記とともに次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT システム権限</li> <li>• パーティションが作成される DB 領域に対する CREATE 権限</li> </ul>
ALTER	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• そのテーブルに対する ALTER 権限</li> <li>• テーブルを所有していること</li> </ul> <p>プライマリキーまたは一意性制約を変更するには、そのテーブルに対する REFERENCES 権限も必要。</p>

句	必要な権限
DROP	<p>制約を持たないカラムを削除するには、次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• ALTER ANY OBJECT システム権限</li> <li>• ALTER ANY TABLE システム権限</li> <li>• 基礎となるテーブルに対する ALTER 権限</li> <li>• 基本となるテーブルを所有していること</li> </ul> <p>ALTER 権限を使用する場合、制約を持つカラムまたはテーブルを削除するには、上記とともに REFERENCES 権限が必要。</p> <p>自分が所有するテーブルのパーティションを削除する場合は、何も必要ない。</p> <p>他のユーザが所有するテーブルのパーティションを削除する場合は、次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• そのテーブルに対する ALTER 権限</li> </ul>
RENAME	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• そのテーブルに対する ALTER 権限</li> <li>• テーブルを所有していること</li> </ul>
MOVE	<p>次のいずれかが必要。</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• システム権限</li> <li>• 基礎となるテーブルに対する ALTER 権限</li> <li>• 基本となるテーブルを所有していること</li> </ul> <p>次のいずれかも必要。</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT システム権限</li> <li>• パーティションの移動先となる DB 領域に対する CREATE 権限</li> </ul>

句	必要な権限
SPLIT PARTITION	自分が所有するテーブルのパーティションの場合は、何も必要ない。 他のユーザが所有するテーブルのパーティションの場合は、次のいずれかが必要。 <ul style="list-style-type: none"> <li>• SELECT ANY TABLE システム権限</li> <li>• テーブルに対する SELECT 権限</li> </ul> 次のいずれかも必要。 <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• そのテーブルに対する ALTER 権限</li> </ul>
MERGE PARTITION、 UNPARTITION	自分が所有するテーブルの場合は、何も必要ない。 他のユーザが所有するテーブルの場合は、次のいずれかが必要。 <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• そのテーブルに対する ALTER 権限</li> </ul>
PARTITION BY	次のいずれかが必要。 <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT システム権限</li> <li>• パーティションが作成される DB 領域に対する CREATE 権限</li> </ul> 次のいずれかも必要。 <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> <li>• そのテーブルに対する ALTER 権限</li> <li>• テーブルを所有していること</li> </ul>
/DISABLE RLV STORE	次のいずれかが必要。 <ul style="list-style-type: none"> <li>• ALTER ANY TABLE システム権限</li> <li>• ALTER ANY OBJECT システム権限</li> </ul>

# 索引

## A

ALTER TABLE 文  
  構文 73  
ALTER VIEW 文  
  RECOMPILE 73

## R

REFERENCES 句 73

## S

SELECT \* 73

## か

カラム  
  変更 73

## く

空間参照系  
  削除 24

作成 24

変更 24

空間測定単位

  削除 25

  作成 25

## て

テーブル

  定義の変更 73

  変更 73

## は

パーティション

  削除 73

パーティションの削除 73

## ひ

ビュー

  変更されたテーブル 73

