UNIT II SPATIAL AND TEMPORAL DATABASES 9

Active Databases Model – Design and Implementation Issues - Temporal Databases - Temporal Querying - Spatial Databases: Spatial Data Types, Spatial Operators and Queries – Spatial Indexing and Mining – Applications -– Mobile Databases: Location and Handoff Management, Mobile Transaction Models – Deductive Databases - Multimedia Databases.

SPATIAL DATABASE

Spatial data is associated with geographic locations such as cities,towns etc. A spatial database is optimized to store and query data representing objects. These are the objects which are defined in a geometric space.

A database is a collection of related information that permits the entry, storage, input, output, and organization of data. A database management system (DBMS) serves as an interface between users and their databases.

A spatial database includes location. It has geometry as points, lines, and polygons. GIS combines spatial data from many sources with many different people. Databases connect users to the GIS database.

For example, a city might have the wastewater division, land records, transportation, and fire departments connected and using datasets from common spatial databases. Let's take a closer look at spatial databases and how we use them in GIS.

Characteristics of Spatial Database

A spatial database system has the following characteristics

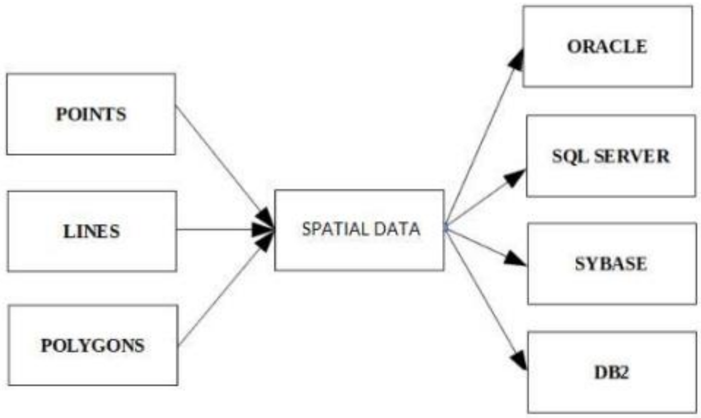
- It is a database system
- It offers spatial data types (SDTs) in its data model and query language.
- It supports spatial data types in its implementation, providing at least spatial indexing and efficient algorithms for spatial join.

Example

A road map is a visualization of geographic information. A road map is a 2-dimensional object which contains points, lines, and polygons that can represent cities, roads, and political boundaries such as states or provinces.

In general, spatial data can be of two types −

- Vector data: This data is represented as discrete points, lines and polygons
- Raster data: This data is represented as a matrix of square cells.



The spatial data in the form of points, lines, polygons etc. is used by many different databases as shown above.

SPATIAL DATA TYPES OVERVIEW

Spatial data represents information about the physical location and shape of geometric objects. These objects can be point locations or more complex objects such as countries, roads, or lakes.

SQL Server supports two spatial data types: the geometry data type and the geography data type. The geometry type represents data in a Euclidean (flat) coordinate system. The geography type represents data in a round-earth coordinate system.

Both data types are implemented as .NET common language runtime (CLR) data types in SQL Server.

There are two types of spatial data. The geometry data type supports planar, or Euclidean (flat-earth), data. The geometry data type both conforms to the Open Geospatial Consortium (OGC)

Simple Features for SQL Specification version 1.1.0 and is compliant with SQL MM (ISO standard). SQL Server also supports the geography data type, which stores ellipsoidal (round-earth) data, such as GPS latitude and longitude coordinates.

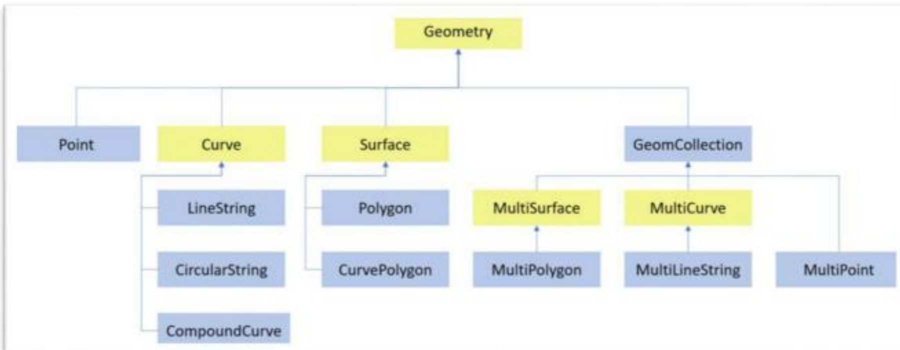
Tip

SQL Server spatial tools is a Microsoft sponsored open-source collection of tools for use with the spatial types in SQL Server. This project provides a set of reusable functions which applications can make use of. These functions may include data conversion routines, new transformations, aggregates, etc.

Spatial data objects

The geometry and geography data types support 16 types of spatial data objects, or instance types. However, only 11 of these instance types are instantiable; you can create and work with these instances (or instantiate them) in a database. These instances derive certain properties from their parent data types.

The figure below shows the geometry hierarchy upon which the geometry and geography data types are based. The instantiable types of geometry and geography are indicated in blue.



Data types for geographic features that can be perceived as forming a single unit; for example, individual residences and isolated lakes.

Data types for geographic features that are made up of multiple units or components; for example, canal systems and groups of islands in a lake.

A data type for geographic features of all kinds.

Data types for single-unit features

Use ST_Point, ST_LineString, and ST_Polygon to store coordinates that define the space occupied by features that can be perceived as forming a single unit.

Data types for multi-unit features

Use ST_MultiPoint, ST_MultiLineString, and ST_MultiPolygon to store coordinates that define spaces occupied by features that are made up of multiple units.

A data type for all features

You can use ST_Geometry when you are not sure which of the other data types to use. The subtypes for geometry and geography types are divided into simple and collection types. Some methods like STNumCurves() work only with simple types.

Simple types are:

Point
LineString
CircularString
CompoundCurve
Polygon

CurvePolygon

Collection types are:

MultiPoint
MultiLineString
MultiPolygon
GeometryCollection

SPATIAL OPERATORS AND QUERIES

1. Spatial operators :

Spatial operators these operators are applied in geometric properties of objects. It is then used in the physical space to capture them and the relation among them. It is also used to perform spatial analysis. Spatial operators are grouped into three categories :

1. Topological operators
2. Projective operators
3. Metric operators

a. Topological operators:

Topological properties do not vary when topological operations are applied, like translation or rotation. Topological operators are hierarchically structured in many levels. The base level offers operators, ability to check for detailed topological relations between regions with a broad boundary. The higher levels offer more abstract operators that allow users to query uncertain spatial data independent of the geometric data model.

Examples – open (region), close (region), and inside (point, loop).

b. Projective operators:

Projective operators, like convex hull are used to establish predicates regarding the concavity convexity of objects. Convexity is a measure of the curvature, or the degree of the curve, in the relationship between bond prices and bond yields. Concavity relates to the rate of change of a function's derivative. A function fff is concave up (or upwards) where the derivative f'f′f, prime is increasing.

Example – Having inside the object's concavity.

c. Metric operators:

Metric operators' task is to provide a more accurate description of the geometry of the object. They are often used to measure the global properties of singular objects, and to measure the relative position of different objects, in terms of distance and direction.

Example – length (of an arc) and distance (of a point to point).

2. Dynamic Spatial Operators:

Dynamic operations change the objects upon which the operators are applied. Create, destroy, and update are the fundamental dynamic operations.

Example – Updation of a spatial object via translate, rotate, scale up or scale down, reflect, and shear.

Operator	Description
SDO_FILTER	Specifies which geometries may interact with a given geometry.
SDO_JOIN	Performs a spatial join based on one or more topological relationships.
SDO_NN	Determines the nearest neighbor geometries to a geometry.
SDO_NN_DISTANCE	Returns the distance of an object returned by the SDO_NN operator.
SDO_POINTINPOLYGON	Takes a set of rows whose first column is a point's x-coordinate value and the second column is a point's y-coordinate value, and returns those rows that are within a specified polygon geometry.
SDO_RELATE	Determines whether or not two geometries interact in a specified way
SDO_WITHIN_DISTANCE	Determines if two geometries are within a specified distance from one another.

Table : lists operators, provided for convenience, that perform an SDO_RELATE operation of a specific mask type.

Spatial Queries

It is a set of spatial conditions characterized by spatial operators that form the basis for the retrieval of spatial information from a spatial database system A request expressed as a combination of spatial conditions (e.g., Euclidean distance from a query point) for extracting specific information from a large amount of spatial data without actually changing these data

A spatial query is a special type of database query supported by geodatabases. The queries differ from SQL queries in several important ways. Two of the most important are that they allow for the use of geometry data types such as points, lines and polygons and that these queries consider the spatial relationship between these geometries

A spatial query uses properties and/or relationships that are of spatial nature and are not explicitly available in the BIM. To process a spatial query the 3D geometry model is analyzed

The requests for the spatial data which requires the use of spatial operations are called Spatial Queries. It can be divided into –

1. Range queries:

It finds all objects of a particular type that are within a given spatial area.

Example – Finds all hospitals within the Siliguri area. A variation of this query is for a given location, find all objects within a particular distance, for example, find all banks within 5 km range.

2. Nearest neighbor queries:

It Finds objects of a particular type which is nearest to a given location.

Example – Finds the nearest police station from the location of the accident.

3. Spatial joins or overlays:

It joins the objects of two types based on spatial conditions, such as the objects which are intersecting or overlapping spatially.

Example – Finds all Dhabas on a National Highway between two cities. It spatially joins township objects and highway objects. Finds all hotels that are within 5 kilometers of a railway station. It spatially joins railway station objects and hotel objects.