An Empirical Study of Spatial Datatypes and their Integration into Query Language

Disha Kawatra
Department of Computer Science and Engineering
Manav Rachna College of Engineering, Faridabad
India

Veepu Uppal
Department of Computer Science and Engineering,
Manav Rachna College of Engineering, Faridabad
India

ABSTRACT

With the expeditious development in Geographic Information System (GIS) and their applications, leads to the increment in the geo-graphical databases that have been developed by different vendors. However, the amalgam of data and accessing is still havoc for the development of GIS. Spatial or geometric data types provide basic abstraction for modeling the geometric structure of objects in space and building the relationships among objects, properties and their operations. Operations on spatial data types include spatial operations on the area of a region, topological relationships checking the relative position of spatial objects to each other like overlap, meet, disjoint, or inside, and cardinal direction relationships like north or southeast.

General Terms

Spatial Database, GIS, geometric data types

Keywords

Keywords: overlap, cardinal direction relationships, disjoint

1. INTRODUCTION

Spatial data provide a fundamental abstraction for modelling the geometric structure of objects in space, their relationships, properties, and operations. Examples: point, line, region; partitions (maps), graphs (networks). A point represents an object for which only its location in space, but not its extent, is relevant. A line is the basic abstraction for facilities for moving through space, or connections in space like roads, cables. A region is the abstraction for something having an extent in 2d-space, e.g. a country, a lake, or a national park. A region may have holes and may also consist of several disjoint pieces. The two most important instances of spatially related collections of objects are partitions and networks. A partition can be viewed as a set of region objects that are required to be disjoint. The adjacency relationship is of particular interest, that is, there exist often pair so region objects with a common boundary. Partitions can be used to represent thematic maps. A network can be viewed as a graph embedded into the plane, consisting of a set of point objects, form ignites nodes, and a set of line objects describing the geometry of the edges.Spatial data often represents 2-dimensional and 3-dimensional data [1].

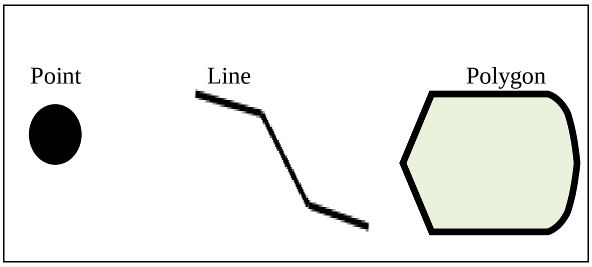

Figure 1.1: The basic three abstraction

A spatial DBMS is a DBMS with additional capabilities for handling spatial data which offers spatial data types in its data model and query language which can structure in space, e.g., point, line, region and has relationships among them, e.g., intersection. It offers spatial data types (SDTs) in its data model and query language. It supports spatial data types in its implementation by providing at least spatial indexing (retrieving objects in particular area without scanning the whole space)[1].

Spatial Database considers modeling, querying and System Architecture.

Modeling considers What Needs to be represented. There are two types of views for modeling the data or what needs to be presented.

Objects in space: - Distinctive entities organized in the expanse which owns its geometric structure.

Space: - Every geometric structure has its own space.

For example, cities, forests or rivers are represented by the first view. The second view is the one of thematic maps describing e.g. land use or the partition of a country into districts. We can reconcile both views to some extent by offering concepts for modeling (i) single objects, and (ii) spatially related collections of objects. For modeling single objects, the fundamental abstractions are point, line, and region.

In **Querying**, the operations of spatial algebraic connected with the DBMS query language. First, data structure and algorithms can be used as the tools or building blocks within different system architecture [1].

In **System Architecture**, the tools are used to integrate to support spatial data types. SQL/MM is the effort to systematize additions for multi-media and application-specific packages in SQL. SQL is described in prolonged way to handle data like texts, stationary images, spatial data, or to perform data mining. The SQL/MM standard uses consistently the prefix ST_ for all tables, views, types, methods and function names. The prefix stood originally for Spatial and Temporal. The SQL/MM Spatial standard defines SQL: 1999 structured user-defined types and associated methods to provide the ability to support such applications [3].

Integrating Geometry into a Query Language

Integrating geometry into a query language has the following three main aspects:
(i) Denoting SDT values as constants in a query and graphical input of such constants.
(ii) Expressing the four classes of fundamental operations for an embedded spatial algebra.
(iii) Describing the presentation of results.

2. SQL TYPE HIERARCHY

SQL/MM Spatial defines several type hierarchies in which most generalized type is called ST_Geometry. It is used to describe the interconnection between the disparate classes and to establish the inheritance rules for the methods working on the instances of the classes and subclasses. All types are used to represent geometric features in the 2-dimensional space.

The SQL Type Hierarchy is shown in “Figure 1”.

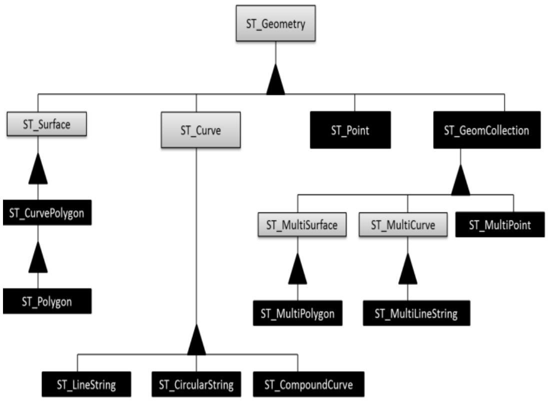

Figure 2.1: SQL Type Hierarchy

ST_Point:-values are 0-dimensional geometries and correspond to an isolated location.

ST_Multipoint:-values stand for a batch of isolated points.

St_Curve:-ST_LineString is described by an arrangement of points, pairs, which defined the reference points of the line string. Circular instead of linear interpolation is used for **ST_CircularStringtype**. A combination of linear and circular strings can be modeled using the ST_CompoundCurve type [2].

ST_Curvepolygon:-stands for such a generalized surface. Subtype ST_Polygon restricts the condition for the rings of the boundary to linear strings.

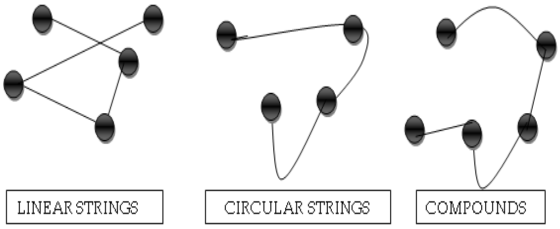

Figure 3: Curves

Consider the following example:

```
CREATE TABLE ocean (
            Ocean_nameVARCHAR(50)
            water_level DOUBLE PRECISION,
            water_depthST_LineString,
            ocean_routeST_LineString,
            cyclone_zonesST_Boundary )
CREATE TABLE  house  (
            Customer1_nameVARCHAR(50)
           house_no VARCHAR(50),
            phone_no VARCHAR(20),
            email_Id VARCHAR(50),
            street1 VARCHAR(50),
            house_plotST_Polygon)
SELECTcustomer1_name,street1,email_Id,phone_no
FROM house AS h, ocean AS o
WHERE h.house_plot.ST_Contains(o.cyclone_zones) = 1
```

ST_Contains is used to check whether one geometry is fully within other geometry. If it returns 1, the spatial relation exists otherwise it returns 0 and spatial relation do not exists.

3. SPATIAL METHODS

Spatial Methods are classified into four categories which are as follows:

- Convert between geometries and external data formats
- Retrieve properties or measures from a geometry
- Compare two geometries with respect to their spatial relation
- Generate new geometries from others.

3.1 Convert between Geometries and other geometries

SQL/MM (Multi-Media) describes three external data format which is used to illustrate the geometry.

- Well-known text representation (WKT)
- Well-known binary representation (WKB)
- Geography markup language (GML)

Each type constructor allows the generation of new geometry from WKT or WKB, numeric spatial reference identifier is used optionally.

3.2 Retrieve Properties

All geometries possess certain properties. for example, the area of a polygon or whether a curve is simple(curve which is not self-intersecting). A set of method was defined to query those properties [2]. Available methods used to retrieve the properties of the geometry are as follows:

- **ST_Boundary**- used to retrieve the boundary of the geometry.
- **ST_Valid**- tests whether the geometry is valid or not i.e. correctly defined.
- **ST_X**- retrieves the x-coordinate of the point
- **ST_Length**- returns the length of the linestring or multi-linestring.

3.3 Comparison between two Geometries

Spatial Queries come to compare the geometries. Comparisons are made on the questions like "what is the distance between two particular buildings" or "which building is in the cyclonezone". Available methods used to compare two geometries are as follows:-

- **ST_Equals**-test the spatial equality of two geometry
- **ST_Disjoint**-test whether two geometries do not intersect
- **ST_Intersects, ST_Crosses, and ST_Overlaps**-test whether the interiors of the geometries intersect.
- **ST_Touches**-test whether two geometries touch at their boundaries, but do not intersect in their interiors
- **ST_Within and ST_Contains**-test whether one geometry is fully within the other [2]

3.4 Generate New Geometries

Generate new geometries method allows to generate new geometries from the already existing geometries. A newly generated geometry can be the result of a set operation on the set of points represented by each geometry, it can be calculated by some algorithm applied to a single geometry [2]. Certain methods used to generate new geometries which are as follows:-

- **ST_Buffer**-generate a buffer at a specific distance around the given geometry.
- **ST_ConvexHull**- compute the convex hull for a geometry.
- **ST_Difference, ST_Intersection and ST_Union**-construct the difference, intersection, or union between the point sets defined by two geometries [2].

4. CONCLUSION

Spatial data types are special data types needed to model geometry and to suitably represent geometric data in database systems. Modeling, Querying and System Architecture are thee step to integrate the spatial data types into database. SQL/MM Spatial defines several type hierarchies to support the spatial data types. Many interesting issues related to spatial database systems could not be included in this survey, for example:

- Spatio-temporal modeling
- spatial objects with imprecise boundaries
- multi-scalemodelling/cartographic generalization
- data lineage (maintaining information about precision, collection method etc. of data)
- spatial reasoning/deductive spatial databases
- performance benchmarks for spatial DBMS

Integrating solutions to such problems with the spatial database technology described here will remain fascinating challenge for database researchers for quite some time.

5. REFERENCES

[1] Ralf Hartmut Güting," An Introduction to Spatial Database Systems", Special Issue on Spatial Database Systems of the VLDB Journal ,Vol. 3, No. 4, October 1994..

[2] Knut Stolze," SQL/MM Spatial: The Standard to Manage Spatial Data in Relational Database Systems", BTW 2003, Leipzig, Feb 2003.

[3] Jim Melton, Andrew Eisenberg,"SQL Multimedia and Application Packages (SQL/MM)", Volume 30 Issue 4, December 2001.Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[4] ShashiShekhar, Jin SoungYoo, "Processing in-route nearest neighbor queries:a comparison of alternative approaches", ACM , New York, NY, USA, 2003.

[5] Markus Schneider,"Spatial Data Types", Springer-Verlag, 2009

[6] SamadiAlinia, M. R. Delava, "Applications of Spatial Data Infrastructure in Disaster Management", May 21, 2009.

[7] Mohammed Abdalla, Hoda M. O. Mokhtar, Mohamed Noureldin, "A Unified Approach For Spatial Data Query" International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.6, November 2013.

[8] PrachiMisraSahoo, "Introduction of Geographic Information System", I.A.S.R.I, New Delhi-11001