to make them geometrically interoperable with datasets of a higher absolute positional accuracy.
To date most PAI programs were implemented as coordinated programs. These programs usually focused on the improvement of a particular dataset and were focused on the individual requirements for geometric interoperability of this dataset with those of a higher positional accuracy. In the future the development of a more generic process to improve the accuracy of geospatial datasets would be beneficial. Particularly the implementation of PAI as a web service, performed on the fly, as the data are being prepared or loaded into a client application, will be a powerful tool in order to make datasets geometrically interoperable.

Cross References

This article has been prepared for information purposes only. It is not designed to constitute definitive advice on the topics covered and any reliance placed on the contents of this article is at the sole risk of the reader.

► Data Infrastructure, Spatial

Recommended Reading

1. Dorling, D., Fairbairn D.: Mapping: ways of representing the world. Longman, London (1997)
2. International Organization for Standardization: ISO 19113 Geographic information – Quality principles. Geneva (2002)
3. Havercroft. M.: Consultation paper: Positional accuracy of large-scale data and products. Ordnance Survey, Southampton (1997)
4. Bray, C., Woodsford, P.: Positional Accuracy Improvement: Impacts of improving the positional accuracy of GI databases. In: EuroSDR official workshop publication. no. 48, Dublin. http://www.eurosdr.net/km_pub/no48/html/positional/positional_index.htm (2004)
5. Roensdorf, C.: PAI2: Achieving Geometric Interoperability of Spatial Data. In: EuroSDR official workshop publication. no. 49, Dublin: http://www.eurosdr.net/km_pub/no49/html/PAI2/index.htm (2005)
6. Hettwer, J., Benning, W.: Restklaffenverteilung mit der Natural-Neighbour-Interpolation. Allg. Vermess.-Nachr. **110**, 122–129 (2003) (in German)
7. Chrisman, N.R.: The Error Component in Spatial Data. In: Maguire, D.J., Goodchild, M.F., Rhind, D. W. (eds.) Geographical Information Systems, Longman, London (1991)
8. Goodchild, M. Gopal, S. (eds.): Accuracy of Spatial Databases. Taylor & Francis, London (1989)
9. Roensdorf, C.: Achieving and Maintaining Interoperability of Spatial Data, FIG working week proceedings, Cairo. http://www.fig.net/pub/cairo/papers/ts_14/ts14_02_ronsdorf.pdf (2004)
10. Straub, B.M., Wiedemann, C.: Towards the update of geodata by automatic object extraction. In: Serpico, S.B (ed.) Image and Signal Processing for Remote Sensing VI. SPIE Proceedings vol. 4170, p. 304. The International Society for Optical Engineering Bellingham, WA (2000) http://ipi216.ipi.uni-hannover.de/html/publikationen/2000/straub/straub_gerke_wiedemann.pdf
11. Saalfeld, A.: Conflation: Automated Map Compilation. US Bureau of Census, SRD Research Report Number: Census/SRD/RR-87/24. http://www.census.gov/srd/papers/pdf/rr87-24.pdf (1997)
12. Kampshoff, S.: First International Workshop on Next Generation 3D City Models. Bonn (2005) http://www.gia.rwth-aachen.de/Service/document_download.php?fileTag=kambonn05
13. Ordnance Survey, Positional Accuracy Improvement website, www.ordnancesurvey.co.uk/PAI
14. MAF/Tiger Accuracy Improvement Program, http://www.census.gov/geo/mod/maftiger.html. This article has been prepared for information purposes only. It is not designed to constitute definitive advice on the topics covered and any reliance placed on the contents of this article is at the sole risk of the reader

Position-Aware Technologies

► Location-Aware Technologies

Positioning

► Location-Based Services: Practices and Products

PostGIS

CHRISTIAN STROBL
German Remote Sensing Data Center (DFD),
German Aerospace Center (DLR), Weßling, Germany

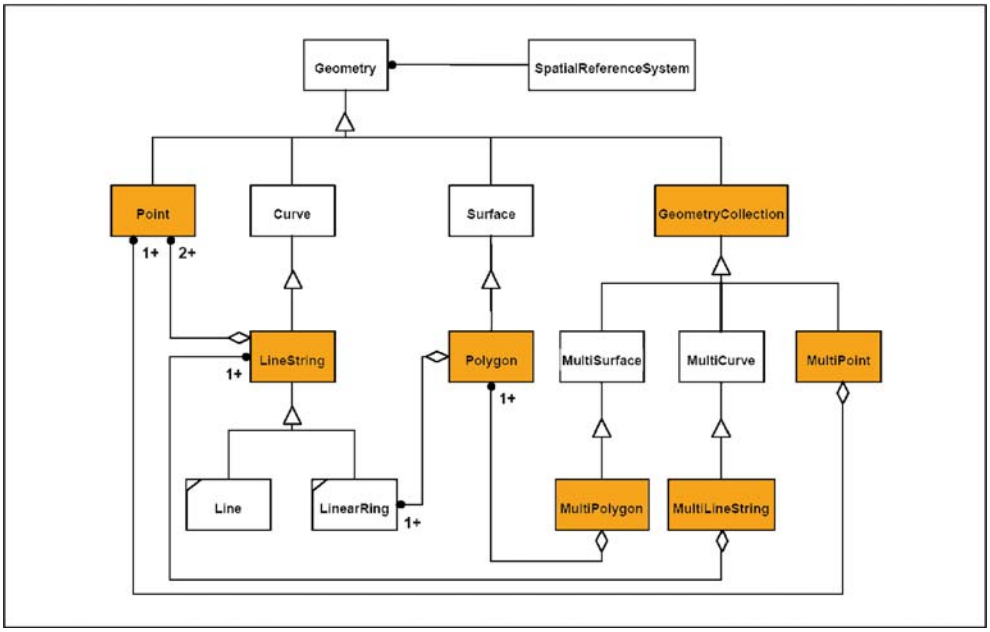
Synonyms

Postgres; OGIS; Spatial DBMS; Public-domain software; Open source; Object-relational; Simple features model; GEOS library; SQL, spatial; R-tree; GiST index

Definition

PostGIS is a spatial database extension for the PostgreSQL (SQL being structured query language) object-relational database. It is certified as a compliant "Simple Features for SQL" database by the Open Geospatial Consortium (OGC).
PostGIS adds geometry data types and spatial functions to the PostgreSQL database. The supported geometry data types are "Points," "LineStrings," "Polygons," "MultiPoints," "MultiLineStrings," "MultiPolygons" and "GeometryCollections". Spatial functions enable the analysis and processing of geographic information systems (GIS) objects. Examples are measurement functions like "Area," "Distance," "Length" and "Perimeter" and spatial operators like "Union," "Difference," "Symmetric Difference" and "Buffer". Topological relationships, like "Equals," "Disjoint," "Intersects," "Touches," "Crosses," "Within," "Contains" and "Overlaps", are processed by the

P



PostGIS, Figure 1 Geometry class hierarchy of the "Simple Features for SQL" specification from the Open Geospatial Consortium. The geometry types supported by PostGIS are gray shaded

Dimensionally Extended Nine-Intersection Model (DE-9IM).

PostGIS and PostgreSQL are open source. PostGIS is released under the GNU General Public License and PostgreSQL is released under the Berkely Software Distribution (BSD) license.

The functionality of PostGIS is comparable to ESRI ArcSDE, Oracle Spatial, and DB II spatial extender.

Historical Background

The first version of PostGIS was released in 2001 by Refractions Research. It is published under the GNU General Public License [1] and development has continued since then. In 2006, PostGIS was certified as a compliant Simple Features for SQL database by the OGC. It uses libraries of other open source projects. The GEOS (Geometry Engine Open Source) library [2] provides most of the operations described by the OGC Simple Features and the proj4 [3] library contributes the projection support.

Refractions Research is located in Victoria, British Columbia, Canada. It is a consulting and product development organization, specializing in spatial and database application development [4].

The history of PostgreSQL begins at the University of California at Berkeley (UCB). PostgreSQL, originally called Postgres, was created at UCB by a computer science professor named Michael Stonebraker. Stonebraker started Postgres in 1986 as a follow-up project to its predecessor Ingres. Stonebraker and his graduate students actively developed Postgres for 8 years. In 1995, two Ph.D. students from Stonebraker's lab, Andrew Yu and Jolly Chen, replaced Postgres' POSTQUEL query language with an extended subset of SQL. They renamed the system to Postgres95. In 1996, Postgres95 departed from academia and started a new life in the open source world under the BSD license [5]. At the same time the database system was given its current name PostgreSQL. PostgreSQL began at version 6.0 (1996); in 2007 the current version is PostgreSQL 8.2 [6].

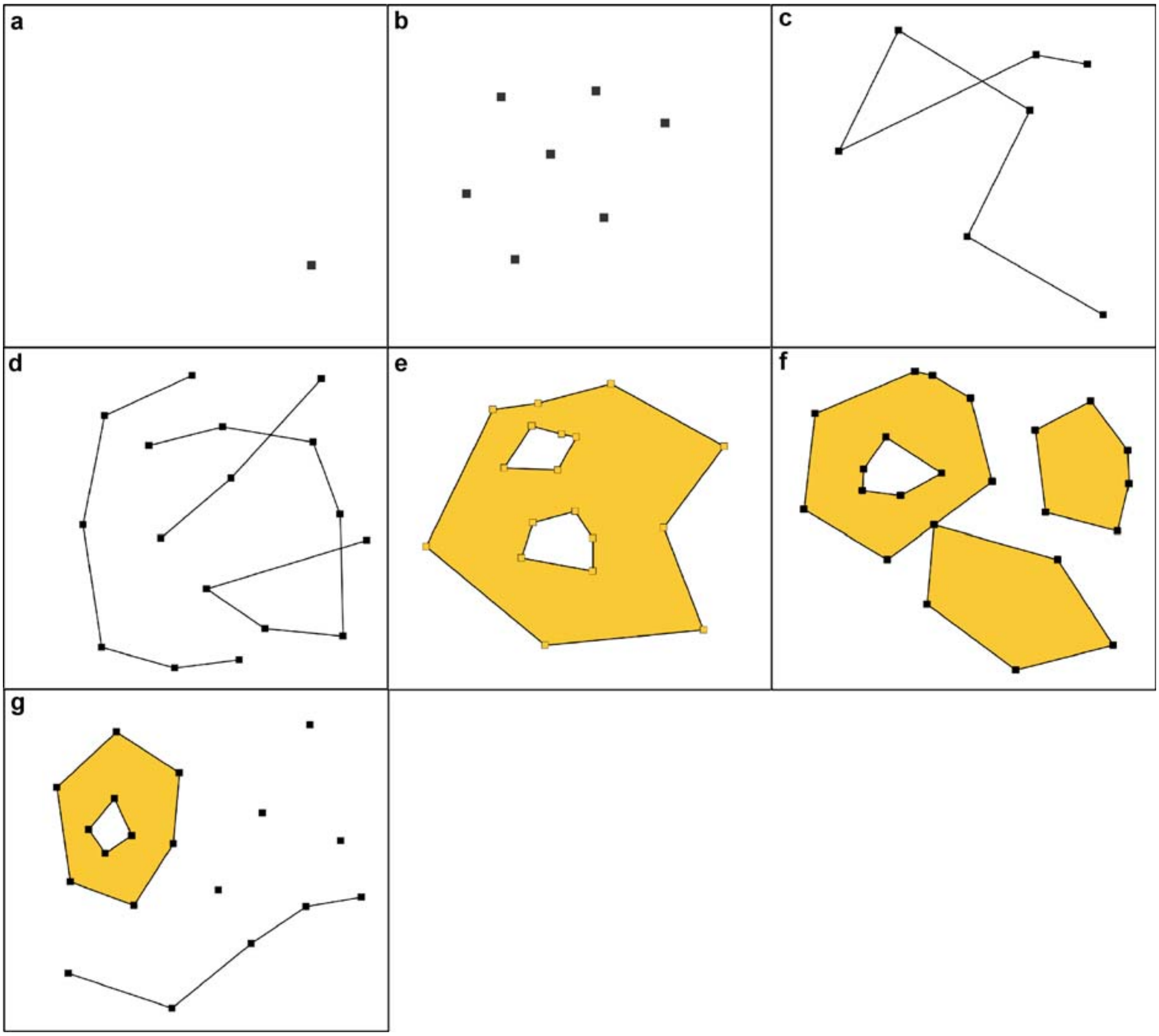
Scientific Fundamentals

Like other spatial databases, PostGIS combines the advantages of classical GIS software, mainly the possibility of spatial analysis, with the advantages of database management systems (DBMS) Such as indexing, transactions and concurrency [7,8].

Simple Features for SQL

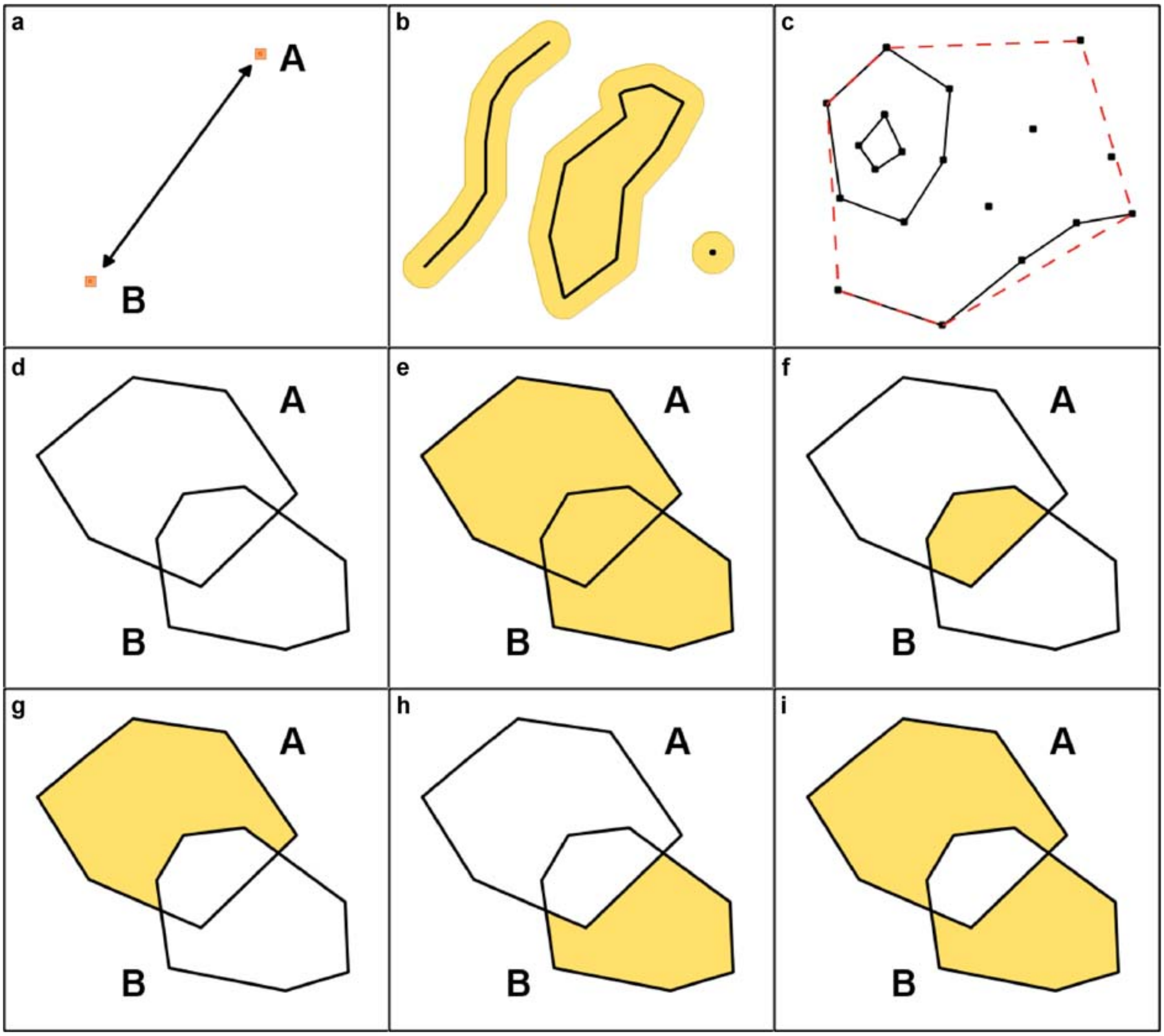
PostGIS follows the Simple Features for SQL specification from the OGC [9]. This implies:

- PostGIS supports the Simple Feature Class Hierarchy according the Open GIS Geometry Model. This includes geometry types for Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, MultiPolygons and GeometryCollections (Fig. 1 and Fig. 2).



PostGIS, Figure 2 Geometry types supported by PostGIS. **a** Point. **b** MultiPoint. **c** LineString. **d** MultiLineString. **e** Polygon. **f** MultiPolygon. **g** GeometryCollection

- PostGIS supports the representation of geometry data as Well Known Text (WKT), Well Known Binary (WKB), as Geography Markup Language (GML) and as Keyhole Markup Language (KML) for Google Earth. Additionally, it supports output as Scalable Vector Graphics (SVG) path geometry.
- PostGIS implements SQL functions that test spatial relationships. These functions include "Equals," "Disjoint," "Intersects," "Touches," "Crosses," "Within," "Contains," "Overlaps" and "Relate". All these operators are based on the DE-9IM [9,10].
- PostGIS implements SQL functions that support spatial analysis. These functions include "Distance," "Buffer," "Convex Hull," "Intersection," "Union," "Difference" and "Symmetric Difference" (Fig. 3).
- PostGIS implements spatial operators for determining geospatial measurements like Area, Distance, Length and Perimeter.
- PostGIS provides information about the geometry type and the spatial reference system. This spatial metadata is stored in the Geometry Columns Metadata View and in the Spatial Reference System Information View according to the Simple Features for SQL specification [9]. Each reference system has a unique identifier called SRID according to the European Petroleum Survey Group (EPSG) code [11].



PostGIS, Figure 3 Spatial functions supported by PostGIS. **a** Distance. **b** Buffer. **c** Convex hull. **e** Union. **f** Intersection. **g** The difference of polygon *A* to polygon *B*. **h** The difference of polygon *B* to polygon *A*. **i** Symmetric difference. **d** The polygons used for the spatial operations of **e**–**i**

Spatial SQL

The implementation of the OGC "Simple Features for SQL" offers GIS new and powerful features for managing, retrieving and analyzing geospatial data. The spatial domain introduces a new set of functions to the SQL Language. The following queries are not complete and give only an elementary review of the potential of the spatial SQL syntax provided by PostGIS:

List the names of all cities which are located inside Bavaria.

```
SELECT city_name
    FROM city a, country b
        WHERE WITHIN (a.geom, b.geom)
        AND b.country_name = 'Bavaria';
```

```
        city_name
----------------------------
Munich
Augsburg
...
```

List the names of all countries which are neighbors to Bavaria.

```
SELECT b.country_name
    FROM country a, country b
        WHERE TOUCHES (a.geom, b.geom)
```

AND a.country_name = 'Bavaria';

```
     country_name
-----------------------------
Thuringen
Baden-Wurttemberg
Hessen
Sachsen
(4 rows)
```

List the names of all cities which are located within 50 km of the river Isar.

SELECT DISTINCT a.city_name
FROM city a, river b
WHERE DISTANCE (a.geom, b.geom) < 50000
AND b.river_name = 'Isar';

```
     city_name
----------------------------
Munich
Passau
...
```

Calculate the area of a buffer of 50,000 m around Munich (see also Fig. 3b).

SELECT AREA (BUFFER(geom,50.000))/ 10000 AS Hectares
FROM city
WHERE city_name = 'Munich';

```
     hectares
------------
780361.288064939
(1 row)
```

List the name, the population and the area of all countries which have an area greater than 3,000,000 ha sorted by the population (in ascending order).

SELECT country_name, pop_admin, AREA(geom)/ 10000 AS Hectares
FROM country
WHERE AREA(geom) > 30000000000
ORDER BY pop_admin;

```
country_name        | pop_admin | hectares
--------------------+-----------+-----------------
Niedersachsen       | 8000909   | 4733454.20332757
Baden-Wurttemberg   | 10717419  | 3621827.63163362
Bayern              | 12469000  | 7029553.1603014
Nordrhein-Westfalen | 18058000  | 3438200.2301504
(4 rows)
```

Show the geometry type of the table cities.

SELECT DISTINCT GEOMETRYTYPE(geom)
FROM city;

```
geometrytype
------------
POINT
(1 row)
```

Show the description of the spatial reference system for the table countries.

SELECT DISTINCT SRID(geom)
FROM country;

```
srid
----
4326
(1 row)
```

With the result of the last query, e. g., the SRID = 4,326, it is possible to get information about the used projection from the table spatial_ref_sys.

SELECT srid, proj4text
FROM spatial_ref_sys
WHERE srid = 4326;

```
srid | proj4text
-----+-------------------------------
4326 | +proj=longlat +ellps=WGS84
               +datum=WGS84 +no_defs
(1 row)
```

Show the point location of Munich as WKT.

SELECT city_name, ASTEXT(geom) AS "Location"
FROM city
WHERE city_name = 'Munich';

```
city_name | Location
----------+------------------------
Munich    | POINT(11.5429545454545
                  48.1409727272727)
(1 row)
```

Show the point location of Munich as GML, transformed to the coordinate system with EPSG code 31464 (Gauß Krüger, Germany, 12th meridian).

SELECT ASGML(TRANSFORM(geom,31464),7)
FROM city
WHERE city_name = 'Munich';

```
asgml
-------------------------------------
<gml:Point srsName="EPSG:31464">
<gml:coordinates>4466089,5333763
                   </gml:coordinates>
```

P

```
<gml:Point>
(1 row)
```

Spatial Join (Query Processing)

PostGIS supports spatial joins. A spatial join is comparable to a standard table join based on a spatial relationship. A standard table join merges two tables into one output result. The join is based on a common key.

SELECT a.city_name, b.country_name
FROM city a, country b
WHERE a.country_name = b.country_name
ORDER BY b.country_name, a.city_name;

A spatial join merges two tables into one output result based on a spatial relationship. For example, the names of the countries are stored in the table country and the names of the cities are stored in the table city. If anybody wants to list the name of the cities and the name of the countries, in which the cities are located, in one table, they have to use a spatial join:

SELECT a.city_name, b.country_name
FROM city a, country b
WHERE WITHIN(a.geom, b.geom)
ORDER BY b.country_name, a.city_name;

Indexing and Query Optimization

PostgreSQL supports compound, unique, partial, and functional indexes, which can use any of its B-tree, R-tree, hash, or Generalized Search Tree (GiST) storage methods. GiST indexing is an advanced system, which provides an interface and framework for developers to add their own indexes. It allows the combination of a lot of different sorting and searching algorithms including B-tree, B+-tree, R-tree, partial sum trees, ranked B+-trees and others [6,12,13,14].

PostGIS indexes are R-tree indexes, implemented on top of the general GiST indexing schema. R-trees organize spatial data into nesting rectangles for fast searching ([4,15], Fig. 4).

With PostgreSQL and PostGIS, several possibilities exist for query optimization. It is possible to choose between a sequential scan and an index scan for attribute data and between a sequential scan and an index scan using the GiST index for geometry data.

For mixed spatial/nonspatial queries it is possible to use the index with the best selectivity to provide high-performance query plans.

The spatial indexes are not used automatically for every spatial request or operator. Because the R-tree index is based on rectangles, spatial indexes are only efficient for bounding box comparisons. In PostGIS, the indexed search is activated by using the "&&" operator, which means "bounding boxes overlap". The following SQL statement shows a short example:

SELECT river_name FROM river
WHERE geom && SETSRID('BOX3D
(11 47, 12 49)'::BOX3D,4326)
AND DISTANCE(geom, GEOMFROMTEXT
('POINT(12.0 48.5)', 4326)) < 1;

The example query demonstrates a characteristic "two-step" approach to spatial processing:

- The first step, the so-called filter step, is the indexed bounding box search, which runs on the whole table (geom && BOX3D).
- The second step is the so-called refinement step. It only operates on the filtered subset using the exact geometries and represents the original query, in this case the distance query. As this query runs only on the subset returned by the filter step the high costs of processing the exact feature geometries are minimized.

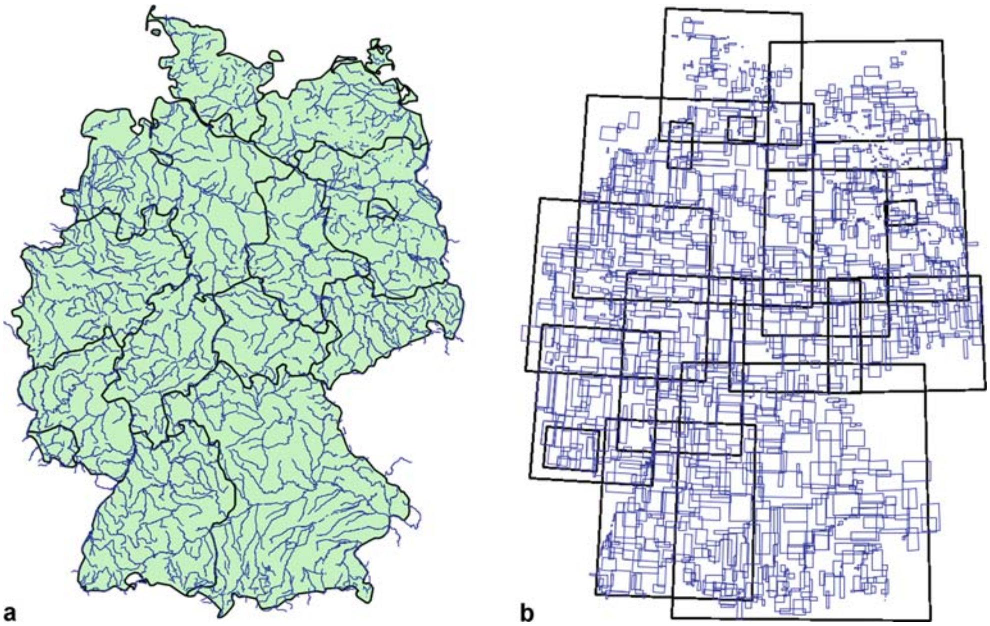
This highly recommended strategy to improve the performance of spatial queries is called the filter-refine paradigm [7].

Key Applications

Spatial data infrastructures (SDIs) facilitate access to geospatial information using a minimum set of standard practices, protocols, and specifications [16]. Every SDI requires a spatial database server and PostGIS represents an open-source- and OGC-compliant solution. Thus PostGIS is supported by many GIS applications, which cover a broad range from server, over workstation and desktop, to internet solutions.

Open Source Software

- deegree: http://www.deegree.org/
- GeoServer: http://geoserver.org/
- GeoTools: http://geotools.codehaus.org/
- GRASS: http://grass.itc.it/
- gvSIG: http://www.gvsig.gva.es/
- MapServer: http://mapserver.gis.umn.edu/
- OGR Simple Feature Library: http://gdal.maptools.org/ogr/
- OpenJUMP: http://openjump.org/wiki/show/HomePage
- Quantum GIS: http://www.qgis.org/
- Thuban: http://thuban.intevation.org/
- uDig: http://udig.refractions.net/
- ...



PostGIS, Figure 4 The bounding boxes that are used for the spatial indexes of the countries and rivers shown in **a** are given in **b**

Proprietary/Closed Software

- ArcGIS (with the Interoperability Extension): http://www.esri.com/
- Cadcorp SIS: http://www.cadcorp.com/
- Feature Manipulation Engine FME: http://www.safe.com/
- Ionic Red Spider: http://www.ionicsoft.com/
- ...

Future Directions

The 1.2.0 release of PostGIS comes with the first support for "curve" types, based on the International Organization for Standardization (ISO) SQL/MM (SQL Multimedia and Application Packages) model for curves. Also initial support for the ISO SQL/MM suite of spatial database functions is implemented [17].

In addition to the ongoing implementation of the ISO SQL/MM standard the PostGIS team works on three-dimensional surface and spline curve support, topology, networks, routing, long transactions and raster integration. The initial groundwork for using PostGIS as an ESRI ArcSDE style interface was also laid in version 1.2. This includes support for most of the ST_* and SE_* spatial SQL functions used by the ArcSDE spatial SQL interfaces.

Cross References

► Data Infrastructure, Spatial
► deegree Free Software
► Dimensionally Extended Nine-Intersection Model (DE-9IM)
► OGC's Open Standards for Geospatial Interoperability
► Oracle Spatial, Geometries
► University of Minnesota (UMN) Map Server

P

Recommended Reading

1. Free Software Foundation (1991) GNU General Public License. http://www.gnu.org/copyleft/gpl.html. Accessed 1 Jan 2007
2. GEOS (2006) Geometry Engine Open Source. http://geos.refractions.net. Accessed 1 Jan 2007
3. PROJ.4 (2000) Cartographic Projections Library. http://www.remotesensing.org/proj. Accessed 1 Jan 2007
4. Refractions Research (2007) PostGIS. http://postgis.refractions.net/. Accessed 1 Jan 2007
5. University of California (Berkeley) (1999) The BSD License. http://www.opensource.org/licenses/bsd-license.php. Accessed 1 Jan 2007
6. PostgreSQL Global Development Group (2007) PostgrSQL. http://www.postgresql.org/. Accessed 1 Jan 2007
7. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall, Upper Saddle River, NJ (2003)
8. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases: With Application to GIS. Morgan Kaufmann, San Fransisco (2002)
9. OGC (ed.): OpenGIS Simple Features Specification for SQL (Revision 1.1). OGC (1999)
10. ISO/TC211: ISO 19107 Geographic Information—Spatial Schema. ISO (2003)
11. OGP Surveying & Positioning Committee (2006) EPSG Geodetic Parameter Dataset v 6.11.2. http://www.epsg.org. Accessed 1 Jan 2007

12. The GiST Indexing Project (1999) GiST: A Generalized Search Tree for Secondary Storage. http://gist.cs.berkeley.edu. Accessed 1 Jan 2007
13. Sigaev, T., Bartunov, O. (2006) GiST for PostgreSQL. http://www.sai.msu.su/~megera/postgres/gist. Accessed 1 Jan 2007
14. Hellerstein, J.M., Naughton, J.F., Pfeffer, A.: Generalized search trees for database systems. In: Proceedings of the 21st International Conference on Very Large Data Bases, Zürich, Switzerland, 11–15 Sept 1995
15. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proceedings of ACM SIGMOD International Conference on Management of Data Boston, MA, 18–21 June 1984
16. Nebert, D.D. (2004) Developing Spatial Data Infrastructures: The SDI Cookbook, Version 2.0. http://www.gsdi.org/pubs/cookbook/cookbookV2.0.pdf. Accessed 12 Oct 2005
17. ISO/IEC: ISO/IEC 13249-3: Information Technology—Database Languages—SQL Multimedia and Application Packages. Part 3: Spatial (2006)

Postgres

► PostGIS

Preference Structure

► Multicriteria Decision Making, Spatial

Prism, Network Time

► Time Geography

Prism, Space-Time

► Time Geography

Privacy

► Cloaking Algorithms for Location Privacy

Privacy and Security Challenges in GIS

BHAVANI THURAISINGHAM, LATIFUR KHAN,
GANESH SUBBIAH, ASHRAFUL ALAM,
MURAT KANTARCIOGLU
Department of Computer Science,
The University of Texas at Dallas, Dallas, TX, USA

Synonyms

Geographic data management

Definition

Geospatial data refers to information about shapes and extent of geographic entities along with their locations on the surface of the earth. This definition, however, is often extended to include any physical or logical entity as long as it exhibits one or more geographic characteristics such as topology of a proposed highway infrastructure or location of a moving vehicle. Geospatial data management pertains to the acquisition, manipulation and dissemination of geospatial data under a set of guidelines. It has numerous applications including counter-terrorism, climate-change detection and space exploration. For example, global warming has been one of the major climate changing events in recent years. The significance of global warming lies in the severe impact that even small climate changes could cause on weather patterns, ecosystems and other activities. Understanding the causes and impacts of global warming is therefore critical. Central to this mission are the thousands of stations capturing vast amounts of geospatially referenced climate and weather data, both on and off the Earth. The data is stored in hundreds of geographically distributed databases, often in different formats. Even more problematic is that the data lack a common semantics, and as a result tends to take on different meanings in different places. These two problems are major impediments to scientists in their ability to coherently and consistently analyze the data, and investigate global trends, make predictions, and so forth.

One way to effectively analyze and detect climate changes is to apply knowledge discovery techniques, also referred to as data mining, for geospatial data sources. If the experts are to systematically process the data in order to answer important scientific and social questions, a coherent representation of the geospatial data related to global warming is needed. The semantic heterogeneity problem is handled by establishing domain ontologies (e. g., emission model, temperature model, sea-level model) to aid in the process of data annotation. A large number of existing environmental parameters can be mapped to geospatial data objects and the remaining ones could be added on gradually.

While the geospatial data related to climate modeling and changes, as well as much of the geospatial data such for counter-terrorism applications such as photographs of building and bridges, are usually publicly available, certain fields may be sensitive to a particular organization. Furthermore, the results of the integration and analysis of the geospatial data may also be sensitive. A recent report by Rand Corporation has stated that geospatial data, even those publicly available, have security needs that must be dealt with [1]. National Oceanic and Atmosher-