Database, Spatial

Spatial databases are the foundation for computer-based applications involving spatially referenced data (i.e. data related to phenomena that have a position, and possibly, a shape, an orientation and a size). Spatial databases can be implemented using various technologies, the most common now being the relational technology. They can have various structures and architectures according to their intended purpose. There are two categories of spatial databases: transactional and analytical. Transactional spatial databases are the most frequent ones; they are often used by geographic information systems (GIS) to facilitate the collection, storage, integrity checking, manipulation and display of the characteristics of spatial phenomena. Analytical databases are more recent, their roots are in the world of statistical analysis and they are central to Business Intelligence (BI) applications. Typical examples include data warehouses and datamarts developed to meet strategic analytical needs. They can comprise multidimensional structures that are called datacubes or hypercubes. When containing spatial data, the datacubes become spatial datacubes.

Spatial databases can store the position, shape, orientation and size of geographic features. Spatial databases can support various types of spatial referencing methods such as 3D geographic coordinates, 2D plan coordinates, 1D linear references (ex. street addresses, road network events, azimuth and distance), 0D point references (ex. place names). Spatial databases accept phenomena that can be points (0D), lines (1D), surfaces (2D) or volumes (3D). Such shapes can be simple, aggregates of simple shapes, optional in some cases, or multiple when more than one shape is required to represent a phenomenon. They can also be static, moving, shrinking, expanding, changing their shape, etc. Spatial databases deal with space in different ways: vector or raster, topological or non-topological, geometry-based or object-based, static or dynamic.

The next section focuses on transactional spatial databases. The third section defines spatial datacubes and related concepts (i.e. dimensions, measures). Then, spatial indexing methods are presented, followed by database architectures concepts and, finally, spatial database design tools and languages.

Transactional spatial databases

A transactional database can be defined as an organized collection of persistent related data used by a group of specific applications. It is typically managed using a particular type of software called a database management system (DBMS) that allows for the definition, entry, storage, processing, modification, querying, diffusion, and protection of data describing various phenomena of interest to the users. It is built to support a large number of small transactions (ex. add, modify, delete) with a large number of concurrent users, to guarantee the integrity of the data and to facilitate updates, especially by keeping data redundancy to a minimum. A spatial database is a database that adds data describing the spatial reference of phenomena. Temporal reference is also possible, leading to spatio-temporal databases when geometric evolutions are supported. Spatial databases can be implemented in a GIS, in a computer-assisted design (CAD) system coupled with a DBMS, in a universal server with a spatial extension, in a spatial engine accessed through an application programming interface (API) and sitting of top of an extended relational database, in a web server with a spatial viewer, etc. These spatial databases can use relational, object-oriented or hybrid structures and they can be organized in very diverse architectures such as centralized and distributed.

The most common transactional approach is the relational approach. It involves concepts such as tables (or relations) made of rows (or tuples) that include data about geographic features, and columns that indicate what the tuple data refer to (identifier, attributes, keys to other features). A transactional database comprises several interlinked tables that are organized to optimize transactions performance (ex. minimize data redundancy, facilitate updates). In spatial databases, some of these tables store geometric data (ex. coordinates, links between lines and polygons). The standard language to define, manipulate and query relational databases is SQL (Structured Query Language). Systems can query spatial databases using spatial extensions to SQL. Figure 1 presents an example of a transactional implementation of a geographic feature with a 0D (point) spatial reference in the DBMS Oracle and a spatially extended SQL query.

[Database_spatial_Figure_1 goes around here]

Figure 1. A 0D geographic feature (i.e. store locations) (a), the implementation of the corresponding table in Oracle (b), and an example of a spatial SQL query (i.e. the creation of a buffer around the points (c).

Spatial datacubes

New types of systems have recently been developed to specifically fulfil decisional needs; they are called analytical systems and are known on the market as Business Intelligence (BI) solutions. These systems, for which the data warehouse is usually a central component, aim to provide a unified view of several dispersed heterogeneous transactional databases in order to efficiently feed decision-support tools.

In the BI world, data warehouses are based on data structures called multidimensional. The term multidimensional was coined in the mid-80s by the community of computer scientists who were involved in the extraction of meaningful information from very large statistical databases (ex. national census). This concept of multidimensionality refers to neither the x, y, z, and t dimensions typically addressed by the GIS community nor to the multiple formats (ex. vector, raster) as considered by some GIS specialists. It refers to the use of a number of analysis themes (i.e. the dimensions) that are cross-referenced for a more in-depth analysis.

The multidimensional approach introduces new concepts, which include dimensions, members, measures, facts and datacubes. The dimensions represent the analysis themes, or the analysis axis (ex. time, products, and sales territories). A dimension contains members (ex. 2006, men shirts, Quebec region) that are organized hierarchically into levels of details (ex. cities, regions, countries). The members at one level (ex. cities) are aggregated to form the members of the next higher level (ex. regions) inside a dimension. Different aggregation formula can be used (ex. sum, average, maximum, count). Different types of dimensions can be defined: temporal, spatial and descriptive (or thematic). The measures (ex. sales, profits) are numerical values analyzed in relation with the different dimensions. The different combinations of dimension members and their resulting measures values represent facts (ex. the sales of men shirts, in 2006, for the Quebec region was $500,000). A set of measures aggregated according to a set of dimensions is called a datacube. A set of spatial and non-spatial measures organized according to a set of spatial and non-spatial dimensions form a spatial datacube.

Three types of spatial dimensions can be defined: non-geometric spatial dimensions, geometric spatial dimensions and mixed spatial dimensions. In the first type of spatial dimension, the spatial reference uses nominal data only (ex. place names) and no coordinates. The geometric spatial dimension comprises, for all dimension members, at all levels of detail, spatially referenced geometric shapes (ex. polygons to represent country boundaries) to allow their dimension members to be visualized and queried on maps. The mixed spatial dimension comprises geometric shapes for only a subset of the levels of details.

Two types of spatial measures can be defined: geometric and numeric. A geometric measure consists of a set of coordinates resulting from geometric operations such as spatial union, spatial merge or spatial intersection. It provides all the geometries representing the spatial objects corresponding to a particular combination of dimension members. Spatial numeric measures are the quantitative values resulting from spatial operators such as calculate surface, distance and number of neighbours.

Spatial datacubes can be used in decision-support tools such as spatial data mining, spatial dashboards or spatial on-line analytical processing (SOLAP).

Spatial indexing methods

In order to facilitate and accelerate the retrieval of spatial information stored in spatial databases, spatial indexing methods are used. These methods aim at reducing the set of objects to be analyzed when processing a spatial data retrieval operation, also called a spatial query. For further acceleration, these methods typically use a simplified geometry of the features, the most commonly used one being the minimum bounding rectangle. Figure 2 presents an example of the minimum bounding rectangle (the dotted line) of a polygon.

[Database_spatial_Figure_2 goes around here]

Figure 2. The minimum bounding rectangle (the dotted line) of a polygon.

Most spatial indexing methods fit into one of these two categories: space-driven structures or data-driven structures. Methods belonging to the first category are based on partitioning the embedding space into cells, independently of the distribution of geographic features. In a two-dimensional space, the grid file, the quadtree, and the space-filling curve are examples of such methods. Methods belonging to the second category are based on partitioning the set of objects and thus adapt to the distribution of these objects. In a two-dimensional space, the R-tree and its variations (R*tree, R+tree) are examples of such methods.

Architectures

A spatial database architecture sometimes refers to the internal layout of data (hierarchical, networked, relational, object-oriented) but nowadays, it also refers to the way it has been partitioned physically. For example, a centralized architecture implies that the database be supported by a unique platform while providing access to several users. In a distributed architecture, the database is divided and each part is supported by a different platform (and the platforms can have different physical locations), the division being based on different criteria (ex. by department, by region, by year). With spatial datacubes, the division can also be based on the granularity (i.e. the level of detail) of the data (ex. national vs local members). In a corporated architecture, a data warehouse may import data directly from several heterogeneous transactional databases, integrate them, store the result and provide access to a homogeneous database.

In a federated architecture, data are partitioned between servers, for example aggregated data can be stored in a data warehouse while other aggregated data (at the same or at a coarser level of detail) are stored in datamarts. Such federated architecture represents a common three tiered architecture for data warehouses. Other architectures include the many variations of the multi-tiered architectures. In the case of spatial analytical systems, the four-tiered architecture, comprising two data warehouses, is often used: the first warehouse stores the integrated data at the level of detail of the source data (because the integration of the detailed spatial data represents an important effort and the result has a value of its own), the second warehouse aggregates these data and is the source for the smaller, highly aggregated spatial datamarts.

Spatial database analysis and design

Formal methods for database analysis and design have been developed in order to improve the efficiency of the database development process and the quality of the results (i.e. to ensure that the resulting database reflects users' needs with regard to content, capabilities and performance). These methods typically rely on models and dictionaries and they help to understand and to describe more precisely the reality of the users, to master the complexity of the problems being addressed, to facilitate the exchange and the validation of ideas, to improve the programming process, and to ease the maintenance of the database. In other words, database models can be seen as thinking tools, communication tools, development tools, and documentation tools.

The formal methods use at least two levels of models, separating the "what" (conceptual models) from the "how" (physical models). This strategy leads to more robust and reusable results. For example, the method called Model Driven Architecture (MDA) proposes three levels of models: the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM). The method called Rational Unified Process (RUP) proposes four levels of models: the domain model, the analysis model, the design model and the implementation model.

Various visual languages and formalisms have been developed for spatial database modeling. It started in the late 1980s with the work related to the Modul-R language (and the supporting Orion software) which evolved into the spatial PVL (Spatial Plug-in for Visual Languages) compatible with different modeling tools, and at the center of Perceptory modeling tool. Since the mid-1990s, similar languages have also appeared such as CONGOO, Geo-ER, Geo-OM, GeoOOA, MADS, POLLEN, Geo-Frame and OMT-G. The earliest ones were based on entity-relationship (ER) concepts but the most recent ones rely on object-oriented (OO) and ontological concepts. In particular, the Unified Modeling Language (UML) has emerged as a standard in the computing community at large and has been widely adopted in the databases and spatial databases communities. As a result, some of the above spatial database modeling languages extend UML to improve the efficiency of spatial database designers. One has also extended UML for spatial datacubes.

Visual modeling tools are also known as CASE (computer-assisted software engineering) tools, for example IBM Rational Rose and Grandite Silverrun. They typically support database schema drawing, content definition, validation, reporting and automatic database code generation. Dedicated CASE tools also exist for spatial database design, the most widely use being Perceptory that extends UML with the Spatial PVL to support the modeling of both transactional spatial databases and spatial datacubes.

Conclusion

This entry defined the two families of spatial databases: the transactional spatial databases and the analytical spatial databases. The first are defined as organized collections of persistent related data used by a group of specific applications. Their analytical counterparts aim to provide a unified view of several dispersed heterogeneous transactional databases in order to efficiently feed decision-support tools. Supporting concepts such as spatial indexing methods, architectures and analysis and design methods have also been presented.

Sonia Rivest and Yvan Bédard

See also Data models, Database design, Database management system, Data warehouse, Exploratory spatial data analysis, Spatial data server, Spatial indexing.

Further readings

Bédard, Y., Merrett, T. & Han, J. (2001). Fundamentals of spatial data warehousing for geographic knowledge discovery. In H. Miller & J. Han (Eds.), *Geographic Data Mining and Knowledge Discovery*. (pp. 53-73). London: Taylor & Francis.

Date, C. J. (2003). *An Introduction to Database Systems, Eighth Edition*. Addison-Wesley.

Manolopoulos, Y., Papadopoulos, A. N. & Vassilakopoulos, M. G. (Eds.). (2004). *Spatial Databases: Technologies, Techniques and Trends*. Idea Group Publishing.

Rafanelli, M. (2003). *Multidimensional Databases: Problems and Solutions*. Idea Group Publishing.

Rigaux, P., Scholl, M. & Voisard, A. (2001). *Spatial Databases: With Application to GIS (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann.

Shekhar, S., & Chawla, S. (2002). *Spatial Databases: A Tour*. Prentice Hall.

Yeung, A. K. W. & Brent Hall, G. (2006). *Spatial Database Systems: Design, Implementation and Project Management*. Springer.