Spatial Databases

A. C. Winstanley, National University of Ireland, Maynooth, Republic of Ireland

Glossary

Database A collection of interrelated information stored by a computer system.

Database Management System (DBMS) A computer system designed to store, organize, retrieve, and combine information.

Domain A classification of data according to their characteristics, for example, integer numbers and Boolean values.

Relational Model A method of structuring data as a set of tuples or records in relations or tables so that the interdependence of data is modeled.

Spatial Data Data that record the location as well as other attributes of objects.

Introduction

Databases are computer systems designed to store information in a systematic way so that their contents can be easily accessed, managed, changed, and augmented. Spatial databases are such systems designed specifically to include data with spatial attributes, such as geographical location, distance, and extent. The software used to manage and query a database is known as a database management system (DBMS). The most prevalent type of database is the relational database, a tabular scheme in which data are defined in terms of simple data types and operations so that it can be reorganized and accessed in a number of different ways. Most spatial DBMSs extend the relational model to include more-complex spatial data-types and operations. All geographic information systems (GISs) use a spatial database with, in addition, functions for presentation, visualization, and analysis of the spatially referenced data.

The Relational Data Model

The relational data model was first defined by E. F. Codd in 1970 and is based round the notion of data being stored in a series of interrelated relations of data items. A relation is a mathematical notion of groups of data items whose order is not significant. They can be represented as a regular table, with each group represented by one tuple or row, the individual data items occupying cells in columns. In a database, each relation/table holds data on one kind of object, each row representing one object of this kind and each cell one attribute. Each cell holds an atomic datum, which cannot be meaningfully subdivided into smaller items. A relational database is a collection of these tables, along with queries, forms, reports, and macros, all stored in a computer program and all of which are interrelated.

As an example, Table 1 shows a relation representing commercial businesses. Each row represents one business, each column an attribute to describe a business, and each cell contains an indivisible data item. All values in each column are of the same simple domain or type, for example, integer number, string, or Boolean. According to the relational model, the order of rows and columns has no significance and relations with similar structures can be combined using set operations (e.g., union and intersection). Each relation has a key, one or more attributes whose value uniquely identifies each row.

A database will typically consist of several relations, each representing a different kind of entity. For example, the business database might also include an owner table containing details of the people who own businesses (Table 2). The key attribute here could be that holding the name of each person. Each row also contains a value from the key attribute of the businesses table, a foreign key, which records the business owned.

The strength of the relational data model is its rigorous foundation in the mathematical notion of relations which has properties proven to make it efficient and flexible. This includes the ability to normalize a set of relations to minimize data redundancy and optimize

Table 1 A database relation representing businesses

Id	*Name*	*Phone*	*City*
1	Conservation Products Ltd.	472846435	London
2	Timecare	547623810	Birmingham
3	Design Ltd.	657347384	Belfast
4	Archive Books	657839210	Dublin
5	Tom Meek Artist	648390184	New York
6	Northern Music	475627439	Liverpool
7	Technologia	484756282	Manchester
8	Gen Supplies	968567595	London
9	Arcare	947465484	London

Table 2 A database relation representing owners of businesses

Id	*Surname*	*First name*	*Busid*
101	Smith	John	9
102	Jones	Alan	2
103	Murphy	Patrick	4

them for querying. Notations can be devised to specify queries as a set of combinations, manipulations, and extractions of values from relations. Mathematical methods, such as relational calculus and relational algebra, form the basis of computer-based notations of which the most common is Structured Query Language (SQL). Queries can be reduced to a series of primitive operations on relations: selection of specific rows of a relation based on particular values of attributes or projection of specific columns from a relation. The most powerful operations consist of joining two relations together based on common values in key fields – the join. For example, to find the name of whoever owns a particular business we can join together the relations in **Tables 1** and **2** using values in the key field, select the row containing the business name, and project out the column containing the owner name. This is expressed in SQL as:

```
SELECT owner.name
FROM business, owner
WHERE business.id = owner.busid and business.name
= Archive Books
```

Queries are the most common way to interrogate a database. They create new, usually temporary, relations or views of a database. Most DBMSs include tools to analyze data and generate reports also. Languages such as SQL also contain operations to alter or add to the database. Typically, selection involved extracting rows based on an attribute being equal to a value or in a range between two values using the standard Boolean operators.

```
SELECT owner.name
FROM business, owner
WHERE business.id = owner.busid and business.turnover
> 100000;
```

Often to speed up querying, the system compiles indexes, sorted according to natural ordering, to access the relevant rows more efficiently. A typical DBMS will also include facilities for multi-user access, versioning and transaction control to ensure the database is updated in a coherent way. The well-founded methodologies for designing and implementing well-structured relational databases, most notably a technique based on entity-relationship modeling, together with its sound mathematical basis, has resulted in the almost universal dominance of relational databases for commercial and scientific applications.

Spatial Relational Databases

A spatial DBMS typically follows the same pattern as relational databases but includes extra data domains as standard which can model spatial attributes such as location, shape, and extent. Also included are spatial operations to allow these attributes to be manipulated in a similar way to the standard domains. What makes spatial domains different is that they can consist of complex data structures which may vary considerably depending on the application. For many applications the most appropriate is the vector model representing spatial attributes of features explicitly as their geometry of points, lines, and areas. A point would be defined using a suitable coordinate system; a line would be defined as an ordered collection of points; an areal feature would consist of an ordered collection of lines that form a closed polygon. Spatial database systems using the vector model have built-in domains that allow users to model spatial features easily. However, many applications need to represent attributes that vary continuously over space, such as temperature, that are more naturally modeled as raster data grids.

Whatever the underlying model, spatial data types usually have no natural linear ordering on which the usual search, selection, and indexing algorithms can work. Also the topological relationships between objects are generally as important as spatial relationships. Typical queries required by applications may include measuring distances and calculating the areas of features. They also include spatial functions such as the creation of new features through the combination of existing ones (e.g., the intersecting area of two polygons or the buffer zone around a feature) and spatial predicates (e.g., 'is there a bus stop within 500 m of this building?').

A spatial DBMS provides facilities that enable these operations that would otherwise be difficult to efficiently program in a conventional database. Typically, these include tools to edit and create geometrical objects and link them to relational tables. In addition, spatial querying is provided which considers the spatial relationship between objects. Typical queries might involve determining the dimensions or centroid of a single object. Others determine whether objects touch, intersect, overlap, or contain one another – a group of operations classified as spatial joins analogous to the natural join using simple data types.

These spatial query terms are sometimes provided through extensions to SQL or by a dedicated spatial query language. Core domains and operations for spatial data have been defined as international standards by the Open Geospatial Consortium (OGC) and have been implemented in query languages such as Spatial SQL and MsSqlSpatial. So, for example, a query to find all the businesses within 10km of 'Arcare' might take the following form:

```
SELECT b1.name
FROM business b1, business b2
WHERE b2.name = 'Arcare' and
                distance(b1.location, b2.location) < 10;
```

Other Boolean spatial predicates include ones to detect adjacency, intersection, and containment between objects.

Spatial queries are efficiently implemented using spatial indexing. Unlike standard data domains that can be indexed alphabetically or numerically, spatial domains require a more complicated scheme that takes into account their two- or three-dimensional nature. Typical schemes include using a superimposed grid system, R-trees based on the bounding rectangles of objects and quad-trees that use recursive divisions of space to produce indices based on object density.

Other Spatial Database Models

Although the relational model is widespread, it is particularly suited for spatial data that uses the vector model of points, lines, and areas. However, not all tasks are amenable to, and not all spatial data come in, this form. Therefore, some spatial databases are based on alternative schemes such as the network or hierarchical models. Such problems include those dealing with topology and those concerning networks of features, such as roads in transport applications. In addition a large amount of data, particularly that used in environmental modeling or captured using remote-sensing, is in the form of raster images or stored as data grids, and the applications that use them lend themselves better to flat-file organization. Many commercial, spatial database systems, such as Oracle Spatial and ESRI ArcGIS, provide facilities for several types of data model to coexist and interact.

One drawback with the relational model is that the data describing an object tends to be distributed over several relations. The object-oriented model combats this through the use of software engineering concepts, such as encapsulation and inheritance. Indeed, these concepts are very good at modeling object geometry and nonspatial attributes. However, this model inhibits issues, such as indexing and query optimization – factors very important with the large quantity of data usual in spatial databases.

See also: Qualitative Spatial Reasoning; Spatial Ontologies.

Further Reading

Burrough, P. A. and McDonnell, R. A. (1998). *Principles of Geographical Information Systems*. Oxford: Oxford University Press.
Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM* 13, 377–387.
Date, C. (2003). *An Introduction to Database Systems*. Boston, MA: Addison Wesley.
Date, C. (2005). *Database in Depth: Relational Theory for Practitioners*. Sebastopol, CA: O'Reilly.
DeMers, M. N. (2000). *Fundamentals of Geographic Information Systems*. New York: Wiley.
Güting, R. H. (1994). An introduction to spatial database systems. *International Journal on Very Large Data Bases* 3, 357–399.
Heywood, I., Cornelius, S. and Carver, S. (1998). *An Introduction to Geographical Information Systems*. London: Longman.
Longley, P. A., Goodchild, M. F., Maguire, D. J. and Rhind, D. W. (2001). *Geographic Information Systems and Science*. New York: Wiley.
Papadias, D., Zhang, D. and Kollios, G. (eds.) (2007). *10th International Symposium, SSTD 2007: Advances in Spatial and Temporal Databases*. Boston, MA: Springer.
Rigaux, P., Scholl, M. and Voisard, A. (2001). *Spatial Databases – With Application to GIS*. Francisco, CA: Morgan Kaufmann.
Ritchie, C. (2002). *Relational Database Principles*. Florence, KY: Int. Cengage Business Press.
Shekhar, S. and Chawla, S. (2003). *Spatial Databases: A Tour*. Saddle River, NJ: Prentice-Hall.
Worboys, M. and Duckham, M. (2004). *GIS a Computing Perspective*. Boca, FL: CRC Press.

Relevant Websites

http://www.opengeospatial.org
Open Geospatial Consortium.