10

Creating and Maintaining Geographic Databases

OVERVIEW

- After people, the database is arguably the most important part of a GIS because of the costs of collection and maintenance, and because the database forms the basis of all queries, analysis, and decision making.
- Today, virtually all large GIS implementations store data in a database management system (DBMS), a specialist piece of software designed to handle multi-user access to an integrated set of data.
- Databases need to be designed with great care, and to be structured and indexed to provide efficient query and transaction performance.
- A comprehensive security and transactional access model is necessary to ensure that multiple users can access the database at the same time.
- On-going maintenance is also an essential, but very resource-intensive, activity.

LEARNING OBJECTIVES

By the end of this chapter students should:

- **Understand the role of database management systems in GIS;**
- **Recognize structured query language (SQL) statements;**
- **Understand the key geographic database data types and functions;**
- **Be familiar with the stages of geographic database design;**
- **Understand the key techniques for structuring geographic information, specifically creating topology and indexing;**
- **Understand the issues associated with multi-user editing and versioning.**

KEY WORDS AND CONCEPTS

DBMS, (RDBMS, ODBMS, ORDBMS), parsers, middleware, object classes, database tables, keys, normal forms, SQL, SQL/MM, database design, indexes, B-tree indexes, grid indexes, quadtree indexes, R-tree indexes, database editing and update, transactions, long transactions, versioning

OVERVIEW

CHAPTER SUMMARY

10.1 Introduction

- A database can be thought of as an integrated set of data on a particular subject.
- Geographic databases are simply databases containing geographic data for a particular area and subject.
- Lists the advantages of the database approach to storing geographic data over traditional file-based datasets including reducing redundancy, decreasing costs, allowing multiple applications, transfer of knowledge, data sharing, security and standards, concurrent users
- Disadvantages include cost, complexity, single user performance decreased
- Describes how to create and maintain geographic databases, and the concepts, tools, and techniques that are available to manage geographic data in databases.

10.2 Database management systems

- A DBMS is a software application designed to organize the efficient and effective storage and access of data.

- Briefly outlines the capabilities of DBMS which include a data model, a data load capability, indexes, a query language, security, controlled update, backup and recovery, database administration tools, applications and APIs
- This list of DBMS capabilities is very attractive to GIS users and so, not surprisingly, virtually all large GIS databases are based on DBMS technology.

10.2.1 Types of DBMS

- Three main types of DBMS are available to GIS users today: relational (RDBMS), object (ODBMS), and object-relational (ORDBMS).
- A relational database comprises a set of tables, each a two-dimensional list (or array) of records containing attributes about the objects under study.
- Object database management systems (ODBMS) were initially designed to address weaknesses of RDBMS, including the inability to store complete objects directly in the database (both object state and behavior), poor performance for many types of geographic query
- ODBMS have not proven to be as commercially successful as some predicted because of the massive installed base of RDBMS. Thus appeared…
- Hybrid object-relational DBMS (ORDBMS) can be thought of as an RDBMS engine with an extensibility framework for handling objects.
- The ideal geographic ORDBMS is one that has been extended to support geographic object types and functions through the addition of a geographic query parser, a geographic query optimizer, a geographic query language, multidimensional indexing services, storage management for large files, long transaction services, replication services

10.2.2 Geographic DBMS extensions

- Two of the commercial DBMS vendors have released spatial database extensions to their standard ORDBMS products
    - IBM – DB2 Spatial Extender and Informix Spatial Datablade
    - Oracle Spatial
    - spatial capabilities in the core of Microsoft SQLServer
    - Opensource DBMS PostgreSQL has also been extended with spatial types and functions (PostGIS).
- None is a complete GIS software system
- Focus is on data storage retrieval and management
- Technical box 10.1 details Oracle Spatial

10.3 Storing data in DBMS tables

- The lowest level of user interaction with a geographic database is usually the *object class* (also called a layer or feature class), which is an organized collection of data on a particular theme
- Object classes are stored in a standard database *table*, a two-dimensional array of rows and columns.
    - Rows contain objects (*instances* of object classes)
    - Columns contain object properties or attributes
- The data stored at individual row, column intersections are usually referred to as values.
- Geographic database tables are distinguished from non-geographic tables by the presence of a geometry column (often called the shape column).
- To save space and improve performance, the actual coordinate values may be stored in a highly compressed binary form.
- Tables are joined together using common row/column values or *keys*.
- Following joins, all tables can be treated as a single table
- Lists Codd's five principles for the efficient and effective design of tables and introduces the concept of *normal forms*
- Normal forms improve the simplicity and stability of a database and reduce redundancy of tables by splitting them into sub-tables that are re-joined at query time
- Notes that large tables common in geographic applications leads to tendency for non-normalized table designs in GIS
- Includes a worked example of normalization of a simple land parcel tax assessment table

10.4 SQL

- The standard database query language adopted by virtually all mainstream databases is SQL (Structured or Standard Query Language: ISO Standard ISO/IEC 9075).
- May be used directly via command line, compiled in a general purpose programming language or via a GUI
- The third major revision of SQL (SQL 3) which came out in 2004 defines spatial types and functions as part of a multi-media extension called *SQL/MM*.
- There are three key types of SQL statements:
- DDL (data definition language) used to create, alter and delete relational database structures

- DML (data manipulation language) used to retrieve and manipulate data
- DCL (data control language) handle authorization and access
- Text briefly walks through simple examples of the first two of these

10.5 Geographic database types and functions

- Working together, ISO and OGC have defined the core geographic types and functions to be used in a DBMS and accessed using the SQL language.
- Figure 10.5 shows the geometry class hierarchy
- There are nine methods for testing spatial relationships between these geometric objects.
- Each takes as input two geometries and evaluates whether the relationship is true or not.
- Figure 10.6 illustrates two examples
- The full set of Boolean operators to test the spatial relationships between geometries is: Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps, Relate
- Seven methods support spatial analysis on these geometries: Distance, Buffer, ConvexHull, Intersection, Union, Difference, SymDifference

10.6 Geographic database design

10.6.1 The database design process

- All GIS and DBMS packages have their own core data model that defines the object types and relationships that can be used in an application and which drive how data types will be implemented and accessed and how more advanced types of feature types and relationships are created
- Database design involves the creation of conceptual, logical, and physical models in the six practical steps shown in Figure 10.9
- The next sections summarize each of these steps and their products

10.6.1.1 Conceptual model

- Model the user's view
- Define objects and their relationships
- Select geographic representation

10.6.1.2 Logical model

- Match to geographic database types
- Organize geographic database structure

10.6.1.3 Physical model

- Define database schema

10.7 Structuring geographic information

10.7.1 Topology creation

- Two database-oriented approaches have emerged in recent years for storing and managing topology: Normalized and Physical.
- Normalized Model focuses on the storage of an arc-node data structure
    - Is said to be normalized because each object is decomposed into individual topological primitives for storage in a database and then subsequent reassembly when a query is posed.
    - Normalized approach advantages are: similarities to the familiar arc-node concept, geometry is only stored once, access can be via an SQL API
    - Normalized approach disadvantages are: query performance suffers, standard referential integrity rules in DBMS have no provision for the complex topological relationships, updates are problematic due to cascading effects
- Physical Model topological primitives are not stored in the database and the entire geometry is stored together for each object.
    - Only other things required to be stored are the specific set of topology rules
    - Topological relationships are then computed on-the-fly whenever they are required by client applications.
    - Requires an external client or middle-tier application for validating topological integrity
- Figures 10.10 and 10.11 illustrate these forms for the same geometry

10.7.2 Indexing

- Geographic databases tend to be very large and geographic queries computationally expensive
- Indexes speed up searching by allowing random instead of sequential access.
- A database index is, conceptually speaking, an ordered list derived from the data in a table.
- Figure 10.12 and the related paragraphs explain a simple example of the standard DBMS one-dimensional B-tree (Balanced Tree) index that is found in most major commercial DBMS.
- Since these 1D indexes are very poor at indexing geographic objects, several geographic indexing techniques have been developed

10.7.2.1 Grid index

- A grid index can be thought of as a regular mesh placed over a layer of geographic objects.

10.7.2.2 Quadtree indexes

- Quadtrees are data structures used for both indexing and compressing geographic database layers, although the discussion here relates only to indexing.
- Several paragraphs and diagrams explain quadtrees

10.7.2.3 R-tree indexes

- R-trees group objects using a rectangular approximation of their location called a minimum bounding rectangle (MBR) or minimum enclosing rectangle (see Box 10.2).
- Groups of point, line, or polygon objects are indexed based on their MBR.
- Technical Box 10.2 explains that a MBR essentially defines the smallest box whose sides are parallel to the axes of the coordinate system that encloses a set of one or more geographic objects.

10.8 Editing and data maintenance

- *Editing* is the process of making changes to a geographic database by adding new objects or changing existing objects as part of data load or database update and maintenance operations.
- A database *update* is any change to the geometry and/or attributes of one or more objects or any change to the database schema.
- Contemporary GIS come equipped with an extensive array of tools for creating and editing geographic object geometries and attributes.

10.9 Multi-user editing of continuous databases

- It is relatively easy to provide multiple users with concurrent read and query access to a continuous shared database, but more difficult to deal with conflicts and avoid potential database corruption when multiple users want write (update) access.

10.9.1 Transactions

- A group of edits to a database is referred to as a *transaction*.
- Many geographic transactions extend to hours, weeks, and months, and are called *long transactions*

10.9.2 Versioning

- Identifies two kinds of versioning

    - *Pessimistic locking* locks out all but one user during an update operation
    - *Optimistic* versioning allows multiple users to update at the same time
- Versioning addresses the locking concurrency problem and supports alternative representations of the same objects in the database.
- Within a versioned database, the different database versions are logical copies of their parents (base tables). Only the modifications are stored in the database
- *Branching* and *merging* occur as the two versions are managed
- Includes an example illustrating how versioning works

10.10 Conclusion

DBMS require a database administrator (DBA) to control database structure and security, and to tune the database to achieve maximum performance.

ESSAY QUESTIONS

1. What are the main advantages and disadvantages of storing geographic data in a DBMS?
2. Is SQL a good language for querying geographic databases?
3. Why are there multiple methods of indexing geographic databases? Compare and contrast at least two indexing methods.
4. What are the main capabilities that are required of a database management system?
5. 'All right in theory, no good in practice'. To what extent does this statement summarize the state of play in object oriented database work?
6. Outline Codd's five principles of good relational database design. Why are geographic table designs often left in an un-normalized form?
7. What are the main conceptual stages in database design, and what is accomplished in each?
8. Outline what is meant by the term minimum bounding rectangle and explain why it can be useful to compute and store it in a GIS database.
9. What do you understand by the physical and normalized models for storing topology in a geographic database and how do these differ?
10. What are the issues that have to be tackled in the management of geographic databases open to simultaneous use by many users?

MULTIPLE CHOICE QUESTIONS (MCQ)

1. What are the two main reasons why a geographic database is a critical part of an operational GIS?
    a. The cost of its creation
    b. The impact it has on analysis
    c. Bringing all the data together
    d. It is large and complex

2. For each of the listed items, state whether it is as good or a bad feature of the use of a database management system:

Item	Good or Bad?
Single user performance	
Maintenance costs	
Application independence	
Data sharing	
Management complexity	
Transfer between applications	
Multi-user capability	
Cost of acquisition	
Maintenance cost	
Data redundancy	

3. Write out the full version of the following acronyms:
    a. ORDBMS ……
    b. RDBMS ……
    c. ODBMS ……
    d. DBMS ……..

4. Which of the two following SELECT queries of a geographic database is likely to be fastest in execution: (a) or (b)?
    a. Select all households living within 100m of a railway line, then from this group select all those who do not have a motor car
    b. Select all households without a motor car, then from this group select all those living within 100m of a railway line

5. Classify each of the following proprietary products as ORDBMS, geographic middleware or RDBMS:

Product	**Nature**
Oracle	
Informix Dynamic Server	
ArcSDE	
IBM DB2	
Geomedia Transaction Server	
Versant	

6. Write out the full version of the following acronyms:
    a. DDL ……
    b. DCL ……
    c. SQL ……

7. Order the following steps into correct sequence in the creation of a geographic database. Write (1) to (6) as appropriate:

Stage	**Order (1)-(6)**
Geographic database structuring	
Geographic representation	
Object and relationship modeling	
Geographic database typing	
User view analysis	
Database schema definition	

8. Complete the verbal equation: OBJECT = STATE + ……?

9. Which of the following principles are NOT among those suggested by Codd as important in all databases?
    a. Only one value per cell
    b. All cells have numbers
    c. All values in a column are about the same subject
    d. Each row is unique
    e. Every table must be normalized
    f. Rows can be in any order

g. Columns can be in any order

10. Which of the following grid dimensions would not allow the generation of a grid quadtree?
    a. 16 by 16
    b. 100 by 100
    c. 256 by 256
    d. 1024 by 1024

CLASS AND INDIVIDUAL ACTIVITIES

1. Identify a geographic database with multiple layers and draw a diagram showing the tables and the relationships between them. Which are the primary keys, and which keys are used to join tables? Does the database have a good relational design?
2. Extract from a topographic map (or section of a topographic map) a pattern of point objects of interest to you. According to the mapping scale, these might be towns, public houses (‘pubs’ or bars), telephone boxes, railway stations and so on, but do not attempt to use more than, say, 15-30. Create a layer of point objects from this source. Using the method outlined in section 10.7.2.2 and Figure 10.14 create the point quadtree for these data.
3. The classic quadtree index is a compact way of storing some raster grids. Find a raster representation of a polygonal object. For simplicity, restrict your map to less than, say, four objects. Now create a raster grid of size 16 by 16 the area and code each cell ‘1’ if it is more than 50% covered by one of these areas. Using the method outlined in the text, create the grid quadtree for these data. How effectively does the quadtree compress these data? How would you scan the quadtree to access any specific datum?
4. Finally, in this sequence of three, find or create a layer consisting of a few area objects. Using the methods outlined in section 10.7.2 create:
    a. A simple grid index
    b. a region quadtree
    c. an R-tree index using the minimum bounding rectangles
5. How do these indexing devices compare?
6. Section 10.5 lists a set of nine Boolean operators to test the spatial relationships between various geometries. Figure 10.6 (a) and (b) illustrates the ‘contains’ and ‘touch’ relationships. Complete this figure by sketching the remaining seven relationships for all possible pairings of point, line and area base geometries. Note that some of the operations will be impossible in some geometries.

7. Find out as much as you can about extensions to SQL that enable basic geographic operations. Make a list of the key additional features and show why each is necessary. A good place to start is R.V. Kothuri, A Godfrind and E. Beinat (2004) *Pro Oracle Spatial*, Apress: Berkeley, CA, but the paper by M. Egenhofer (1992) Why not SQL? *International Journal of Geographical Information Systems*, 6:71-85 is a less daunting.
8. Figure 10.7 illustrates four geometric spatial analysis operations (buffer, convex hull, intersection and difference) for each geometric type. In each of the twelve illustrated cases suggest how the operation might be used in a real analysis.
9. Following the sequence illustrated in Figure 10.3, and Section 10.6, devise a database structure to store information about a large collection of landscape photographs to include subject, a classification of the subject, place, locational co-ordinates, direction of camera and related information. Having devised the structure, write out SQL commands to CREATE the table and a selection of possible queries involving joined.
10. Frantisek Brabec and Hanan Samet at the University of Maryland have devised a wonderfully comprehensive website with interactive demonstrations of numerous spatial indexing techniques and associated algorithms for deletion and insertion of new data. The applets require the Java Development Kit 1.1 and are based on algorithms published in Samet's classic books: H. Samet (1990) *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS,* Addison-Wesley, Reading, MA, and (1990) *The Design and Analysis of Spatial Data Structures* , Addison-Wesley, Reading, MA.Visit the website at www.cs.umd.edu/~brabec/quadtree and use this resource both to cement and to extend your knowledge of indexing. Pay particular attention to the algorithms for editing the indexes.
11. A key issue in almost all this chapter has been industry attempts to handle the geometric/spatial components of geospatial data using standard database technologies. This has obvious commercial and related advantages, but is arguably not what practitioners of the science of handling such data need if progress is to be made in, for example, handling time varying or three- and more dimensional representations. Organize a debate around the proposal that 'This house believes that use of standard relational database technology is a major impediment to research in representing geographic information'. Use the text for arguments against the motion and for arguments for it almost every essay in the book by Fisher, P.F. and D.J. Unwin (eds, 2005) *Re-presenting GIS*, Wiley: Chichester. The readings also list papers by Egenhofer et al (1999) and an early work by Worboys et al (1990) that have relevant materials. It will also help the 'for' case if the debating team visits websites that describe experimental GIS, such as DOGIS, at www.cse.ohio-state.edu/~prasun/publications/conf/dogis.pdf. There are others.

FURTHER READING

Date C.J. 2003 *Introduction to Database Systems* (8th edn). Reading, MA: Addison-Wesley.

Egenhofer, M., Glasgow, J., Gunther, O., Herring, J.R. and D. J. Peuquet 1999 Progress in computational methods for representing geographic objects. *International Journal of Geographical Information Science*, 13: 775-796 (available at www.spatial.maine.edu/~max/PiCM.pdf)

Is a very good overview of database issues in GIS

Hoel E., Menon S. and Morehouse S. 2003 'Building a robust relational implementation of topology.' In Hadzilacos T., Manolopoulos Y., Roddick J.F. and Theodoridis Y. (eds) *Advances in Spatial and Temporal Databases. Proceedings of 8th International Symposium, SSTD 2003 Lecture Notes in Computer Science*, Vol. 2750.

OGC 1999 OpenGIS simple features specification for SQL, Revision 1.1. Available at www.opengis.org

Samet H. 1990 *The Design and Analysis of Spatial Data Structures*. Reading, MA: Addison-Wesley.

Worboys M.F. and Duckham M. 2004 *GIS: A Computing Perspective* (2nd edn). Boca Raton, FL: CRC Press.

Worboys, M.F., Hearnshaw, H.M. and D.J. Maguire 1990 Object-oriented data modelling for spatial databases. *International Journal of Geographical Information Systems*, 4(4): 369-383

Very much the manifesto in favor of object orientation written by a team from UK's ESRC Regional Research Laboratory.

Zeiler M. 1999 *Modeling our World: The ESRI Guide to Geodatabase Design*. Redlands, CA: ESRI Press.

RELATED READING

Longley P.A., Goodchild M.F., Maguire D.J. and Rhind D.W. (eds) 2005 *Geographical Information Systems: Principles, Techniques, Management and Applications* (abridged edition). Hoboken, NJ: Wiley.

27. Spatial access methods, P van Oosterom, pp. 385-400

Maguire D.J., Goodchild M.F. and Rhind D.W. (eds) 1991 *Geographical Information Systems: Principles and Applications*. Harlow, UK: Longman (text available online www.wiley.co.uk/gis/volumes.html).

18. Database management systems, R G Healey, pp. 251-267

ONLINE RESOURCES

ESRI Virtual Campus course, *Turning Data into Information* by Paul Longley, Michael Goodchild, David Maguire, and David Rhind (campus.esri.com)

Module 3: Query and Measurement

Section 10.4, Module 3: Query and Measurement

Unit: Advanced queries,

Sub-unit: Tabular queries

NCGIA Core Curriculum in GIScience, 2000 (www.ncgia.ucsb.edu/giscc)

3.1. Making it work (136), Hugh Calkins and others

NCGIA Core Curriculum in GIS, 1990 (www.ncgia.ucsb.edu/pubs/core.html)

60. System planning overview

66. Database creation

67. Implementation issues

68. Implementation strategies for large organizations