UC Santa Barbara

Core Curriculum-Geographic Information Systems (1990)

Title

Unit 43 - Database Concepts I

Permalink

https://escholarship.org/uc/item/7gn5d8zm

Authors

Unit 43, CC in GIS

White, Gerald

Publication Date

1990

Peer reviewed

UNIT 43 - DATABASE CONCEPTS I

UNIT 43 - DATABASE CONCEPTS I

Compiled with assistance from Gerald White, California State University, Sacramento

UNIT 43 - DATABASE CONCEPTS I

Compiled with assistance from Gerald White, California State University, Sacramento

A. INTRODUCTION

- very early attempts to build GIS began from scratch, using very limited tools like operating systems and compilers

- more recently, GIS have been built around existing database management systems (DBMS)
    - purchase or lease of the DBMS is a major part of the system's software cost
    - the DBMS handles many functions which would otherwise have to be programmed into the GIS

- any DBMS makes assumptions about the data which it handles
    - to make effective use of a DBMS it is necessary to fit those assumptions
    - certain types of DBMS are more suitable for GIS than others because their assumptions fit spatial data better

Two ways to use DBMS within a GIS

1. Total DBMS solution
    - all data are accessed through the DBMS, so must fit the assumptions imposed by the DBMS designer

2. Mixed solution
    - some data (usually attribute tables and relationships) are accessed through the DBMS because they fit the model well
    - some data (usually locational) are accessed directly because they do not fit the DBMS model

GIS as a database problem

- some areas of application, notably facilities management:
    - deal with very large volumes of data
    - often have a DBMS solution installed before the GIS is considered

- the GIS adds geographical access to existing methods of search and query

- such systems require very fast response to a limited number of queries, little analysis

- in these areas it is often said that GIS is a "database problem" rather than an algorithm, analysis, data input or data display problem

B. CONCEPTS IN DATABASE SYSTEMS

Definition

- a database is a collection of non-redundant data which can be shared by different application systems

stresses the importance of multiple applications, data sharing
- ◦ the spatial database becomes a common resource for an agency

- implies separation of physical storage from use of the data by an application program, i.e. program/data independence
    - ◦ the user or programmer or application specialist need not know the details of how the data are stored
    - ◦ such details are "transparent to the user"

- changes can be made to data without affecting other components of the system. e.g.
    - ◦ change format of data items (real to integer, arithmetic operations)
    - ◦ change file structure (reorganize data internally or change mode of access)
    - ◦ relocate from one device to another, e.g. from optical to magnetic storage, from tape to disk

Advantages of a database approach

- reduction in data redundancy
    - ◦ shared rather than independent databases
        - ▪ reduces problem of inconsistencies in stored information, e.g. different addresses in different departments for the same customer

- maintenance of data integrity and quality

- data are self-documented or self-descriptive
    - ◦ information on the meaning or interpretation of the data can be stored in the database, e.g. names of items, metadata

- avoidance of inconsistencies
    - ◦ data must follow prescribed models, rules, standards

- reduced cost of software development
    - ◦ many fundamental operations taken care of, however DBMS software can be expensive to install and maintain

- security restrictions
    - ◦ database includes security tools to control access, particularly for writing

Views of the database

overhead - Views of the database

- the database can present different views of itself to users, programmers
    - ◦ these are built and maintained by the database administrator (DBA)

- the internal data representation (internal view) is normally not seen by the user or applications programmer

- the conceptual view or conceptual schema is the primary means by which the DBA

builds and manages the database

- the DBMS can present multiple views of the conceptual schema to programmers and users, depending on the application
  - these are called external views or schemas

overhead - Water district database

C. DATABASE MANAGEMENT SYSTEMS

Components

Data types
- includes:
  - integer (whole numbers only)
  - real (decimal)
  - character (alphabetic and numeric characters)
  - date
- more advanced systems may include pictures and images as data types
  - e.g. a database of buildings for the fire department which stores a picture as well as address, number of floors, etc.

Standard operations
  - e.g. sort, delete, edit, select records

Data definition language (DDL)
- the language used to describe the contents of the database
  - e.g. attribute names, data types - "metadata"

Data manipulation and query language
- the language used to form commands for input, edit, analysis, output, reformatting etc.

- some degree of standardization has been achieved with SQL (Standard Query Language)

Programming tools
- besides commands and queries, the database should be accessible directly from application programs through e.g. subroutine calls

File structures
- the internal structures used to organize the data

Types of database systems

- several models for databases:
  - tabular ("flat file") - data in a single table
  - hierarchical
  - network

- relational

- the hierarchical, network and relational models all try to deal with the same problem with tabular data:
    - inability to deal with more than one type of object, or with relationships between objects
    - e.g. database may need to handle information on aircraft, crew, flights and passengers - four types of records with different attributes, but with relationships between them (e.g. "is booked on" between passenger and flight)

- database systems originated in the late 1950s and early 1960s largely by research and development of IBM Corporation

- most developments were responses to needs of business, military, government and educational institutions - complex organizations with complex data and information needs

- trend through time has been increasing separation between the user and the physical representation of the data - increasing "transparency"

D. HIERARCHICAL MODEL

- early 1960s, IBM saw business world organizing data in the form of a hierarchy

- rather than one record type (flat file), a business has to deal with several types which are hierarchically related to each other
    - e.g. company has several departments, each with attributes: name of director, number of staff, address

        - each department requires several parts to make its product, with attributes: part number, number in stock
        - each part may have several suppliers, with attributes: address, price

    diagram

- certain types of geographical data may fit the hierarchical model well
    - e.g. Census data organized by state, within state by city, within city by census tract

    diagram

- the database keeps track of the different record types, their attributes, and the hierarchical relationships between them

- the attribute which assigns records to levels in the database structure is called the key (e.g. is record a department, part or supplier?)

Summary of features

a set of record "types"
- e.g. supplier record type, department record type, part record type

- a set of links connecting all record types in one data structure diagram (tree)

- at most one link between two record types, hence links need not be named
  - for every record, there is only one parent record at the next level up in the tree
    - e.g. every county has exactly one state, every part has exactly one department

- no connections between occurrences of the same record type
  - cannot go between records at the same level unless they share the same parent

    diagram

Advantages and disadvantages

- data must possess a tree structure
  - tree structure is natural for geographical data

- data access is easy via the key attribute, but difficult for other attributes
  - in the business case, easy to find record given its type (department, part or supplier)
  - in the geographical case, easy to find record given its geographical level (state, county, city, census tract), but difficult to find it given any other attribute
    - e.g. find the records with population 5,000 or less

- tree structure is inflexible
  - cannot define new linkages between records once the tree is established
    - e.g. in the geographical case, new relationships between objects
  - cannot define linkages laterally or diagonally in the tree, only vertically
  - the only geographical relationships which can be coded easily are "is contained in" or "belongs to"

- DBMSs based on the hierarchical model (e.g. System 2000) have often been used to store spatial data, but have not been very successful as bases for GIS

E. NETWORK MODEL

- developed in mid 1960s as part of work of CODASYL (Conference on Data Systems Languages) which proposed programming language COBOL (1966) and then network model (1971)
  - other aspects of database systems also proposed at this time include database administrator, data security, audit trail

- objective of network model is to separate data structure from physical storage, eliminate unnecessary duplication of data with associated errors and costs

- uses concept of a data definition language, data manipulation language

- uses concept of m:n linkages or relationships
  - an owner record can have many member records
  - a member record can have several owners
    - hierarchical model allows only 1:n

- example of a network database
  - a hospital database has three record types:
    - patient: name, date of admission, etc.
    - doctor: name, etc.
    - ward: number of beds, name of staff nurse, etc.
  - need to link patients to doctor, also to ward
  - doctor record can own many patient records
  - patient record can be owned by both doctor and ward records

- network DBMSs include methods for building and redefining linkages, e.g. when patient is assigned to ward

Restrictions

- links between records of the same type are not allowed

- while a record can be owned by several records of different types, it cannot be owned by more than one record of the same type (patient can have only one doctor, only one ward)

Summary

- the network model has greater flexibility than the hierarchical model for handling complex spatial relationships

- it has not had widespread use as a basis for GIS because of the greater flexibility of the relational model

F. RELATIONAL MODEL

- the most popular DBMS model for GIS
  - the INFO in ARC/INFO
  - EMPRESS in System/9
  - several GIS use ORACLE
  - several PC-based GIS use DBase III

- flexible approach to linkages between records comes closest to modeling the complexity of spatial relationships between objects

- proposed by IBM researcher E.F. Codd in 1970

- more of a concept than a data structure

  - internal architecture varies substantially from one RDBMS to another

Terminology

- each record has a set of attributes
  - the range of possible values (domain) is defined for each attribute

- records of each type form a table or relation
  - each row is a record or tuple
  - each column is an attribute

- note the potential confusion - a "relation" is a table of records, not a linkage between records

- the degree of a relation is the number of attributes in the table
  - 1 attribute is a unary relation
  - 2 attributes is a binary relation
  - n attributes is an n-ary relation

Examples of relations

unary: COURSES(SUBJECT) binary: PERSONS(NAME,ADDRESS) OWNER(PERSON NAME,HOUSE ADDRESS) ternary: HOUSES(ADDRESS,PRICE,SIZE)

Keys

- a key of a relation is a subset of attributes with the following properties:
  - unique identification
    - the value of the key is unique for each tuple
  - nonredundancy
    - no attribute in the key can be discarded without destroying the key's uniqueness
  - e.g. phone number is a unique key in a phone directory
    - in the normal phone directory the key attributes are last name, first name, street address
    - if street address is dropped from this key, the key is no longer unique (many Smith, John's)

- a prime attribute of a relation is an attribute which participates in at least one key
  - all other attributes are non-prime

Normalization

- concerned with finding the simplest structure for a given set of data
  - deals with dependence between attributes
  - avoids loss of general information when records are inserted or deleted

  overhead - Normalization

- consider the first relation (prime attribute underlined):

this is not normalized since PRICE is uniquely determined by STYLE

- ○ problems of insertion and deletion anomalies arise
    - ▪ the relationship between ranch and 50000 is lost when the last of the ranch records is deleted
    - ▪ a new relationship (triplex costing 75000) must be inserted when the first triplex record occurs

- consider the second relation:
    - ○ here there are two relations instead of one
        - ▪ one to establish style for each builder
        - ▪ the other price for each style

- several formal types of normalization have been defined - this example illustrates third normal form (3NF), which removes dependence between non-prime attributes

- although normalization produces a consistent and logical structure, it has a cost in increased storage requirements
    - ○ some GIS database administrators avoid full normalization for this reason

- a relational join is the reverse of this normalization process, where the two relations HOMES2 and COST are combined to form HOMES1

Advantages and disadvantages

- the most flexible of the database models

- no obvious match of implementation to model - model is the user's view, not the way the data is organized internally

- is the basis of an area of formal mathematical theory

- most RDBMS data manipulation languages require the user to know the contents of relations, but allow access from one relation to another through common attributes

    Example: Given two relations: PROPERTY(ADDRESS,VALUE,COUNTY_ID) COUNTY(COUNTY ID,NAME,TAX_RATE)

- to answer the query "what are the taxes on property x" the user would:
    - ○ retrieve the property record
    - ○ link the property and county records through the common attribute COUNTY_ID
    - ○ compute the taxes by multiplying VALUE from the property tuple with TAX_RATE from the linked county tuple

REFERENCES

Standard database texts:

Date, G.J., 1987. An Introduction to Database Systems, Addison-Wesley, Reading, MA.

Howe, D.R., 1983. Data Analysis for Data Base Design, Arnold, London.

Kent, W., 1983. "A simple guide to five normal forms in relational database theory," Communications of the Association for Computing Machinery 26:120.

Tsichritzis, D.C. and F.H. Lochovsky, 1977, Database Management Systems, Academic Press, New York.

The relational model in GIS:

van Roessel, J.W., 1987. "Design of a spatial data structure using the relational normal forms," International Journal of Geographical Information Systems 1:33-50.

EXAM AND DISCUSSION QUESTIONS

1. Compare the four database models (flat file, hierarchical, network and relational) as bases for GIS. What particular features of the relational model account for its popularity?

2. Polygon overlay has been called a spatial analog of a relational join. Do you agree?

3. Summarize the arguments against organizing spatial databases as flat files.

4. Why do you think the term "relation" was chosen for a table of attributes in the relational model?

Last Updated: August 30, 1997.

UNIT 43 IMAGES