Database Concepts

presented by:
Tim Haithcoat
University of Missouri
Columbia

Introduction

- Very early attempts to build GIS began from scratch, using limited tools like operating systems & 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

Databases for Spatial Data

- Many different data types are encountered in geographical data
  - examples: pictures, words, coordinates, complex objects
- Very few database systems have been able to handle textual data
  - Example: descriptions of soils in the legend of a soil map can run to hundreds of words
  - Example: descriptions are as important as numerical data in defining property lines in surveying - “metes and bounds” descriptions

Databases for Spatial Data (continued)

- Variable length records are needed, often not handled well by standard systems
  - Example: number of coordinates in a line can vary
  - This is the primary reason why some GIS designers have chosen not to use standard database solutions for coordinate data, only for attribute tables

Databases for Spatial Data (continued)

- Standard database systems assume the order of records is not meaningful
  - In geographical data the positions of objects establish an implied order which is important in many operations
    - Often need to work with objects that are adjacent in space, thus it helps to have these objects adjacent or close in the database
    - Is a problem with standard database systems since they do not allow linkages between objects in the same record type (class)

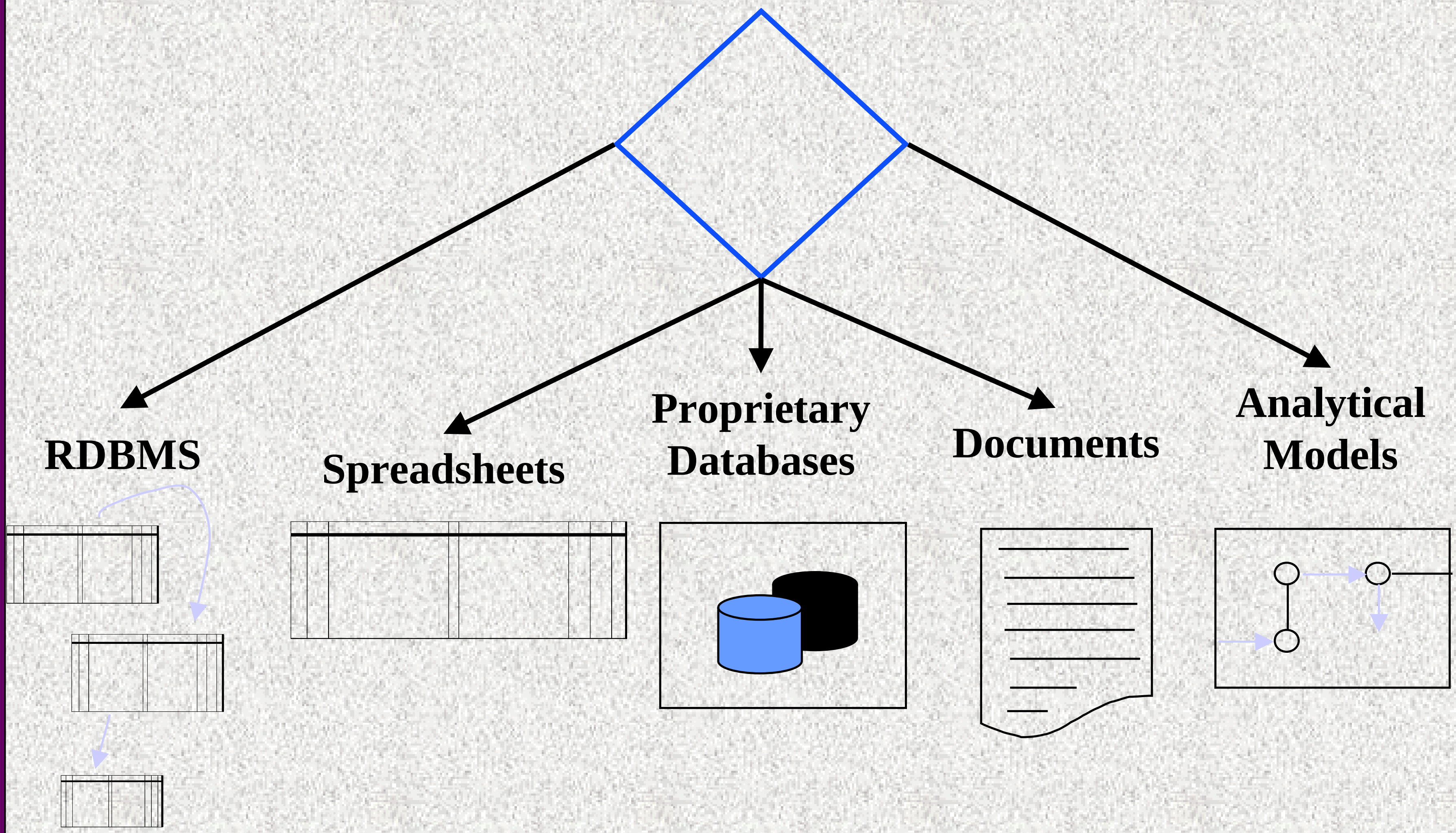
Databases for Spatial Data (continued)

- There are so many possible relationships between spatial objects, that not all can be stored explicitly
  - However, some relationships must be stored explicitly as they cannot be computed from the geometry of the objects
    - Example: existence of grade separation
- The integrity rules of geographical data are too complex
  - Example: the arcs forming a polygon must link into a complete boundary
  - Example: lines cannot cross without forming a node

Two ways to use DBMS within a GIS:

- Total DBMS solution
  - all data are accessed through the DBMS, so must fit the assumptions imposed by the DBMS designer
- 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 Gateway to External Data



GIS as a Database Problem

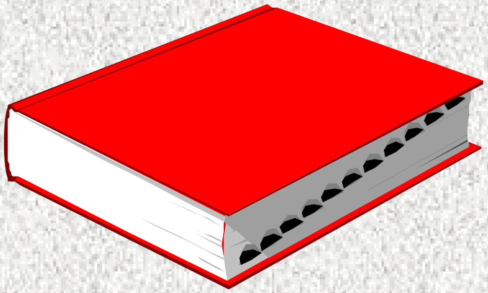
- Some areas of application, notable facilities management:
  - deal with very large volumes of data
  - often have a DBMS solution installed before the GIS is considered
- The GIS adds geographical access to existing methods of search and query
- Such systems require very fast response to a limited 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

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"

Definition (continued)

- 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

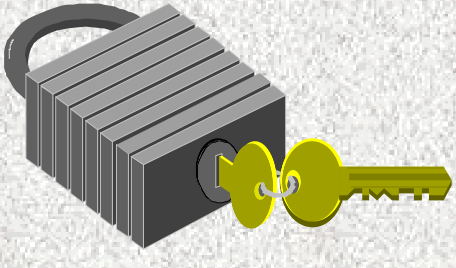


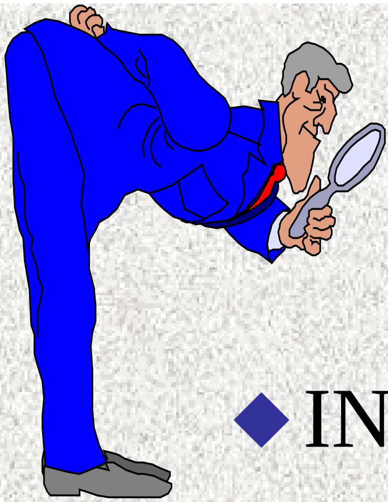
Advantages of a Database Approach

- 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

Advantages of a Database Approach (continued)

- 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



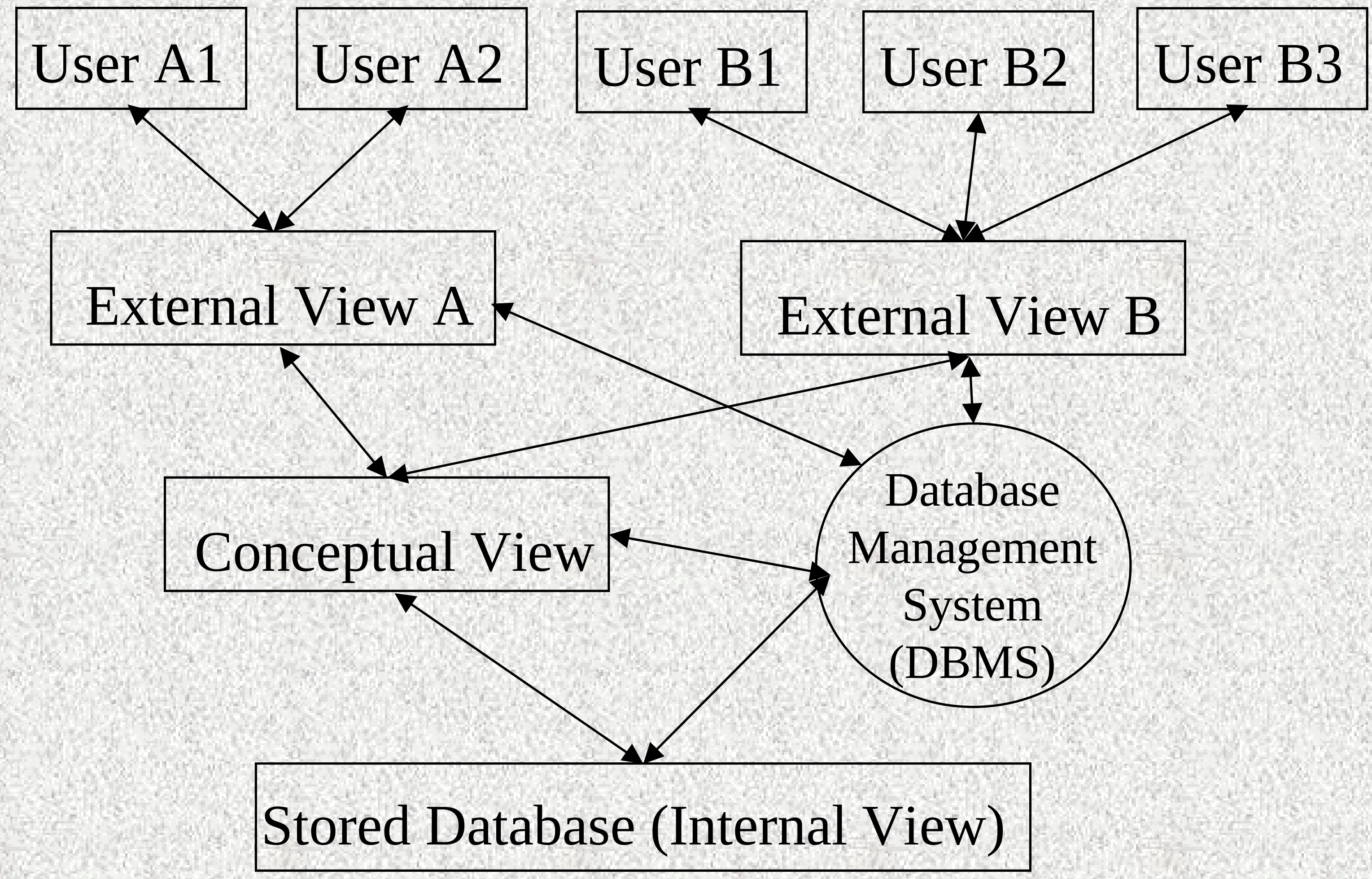


Views of the Database

- INTERNAL VIEW
  - Normally not seen by the user or applications developer
- CONCEPTUAL VIEW
  - Primary means by which the database administrator builds and manages the database
- EXTERNAL VIEW (or Schemas)
  - what the user or programmer sees - can be different to different users and applications

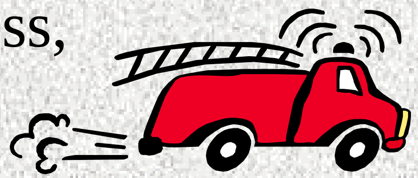
Views of the Database

Adapted from: Date, G.J. 1987. An Introduction to Database Systems, Addison-Wesley. Reading, MA, p. 32



Database Management Systems: *Components*

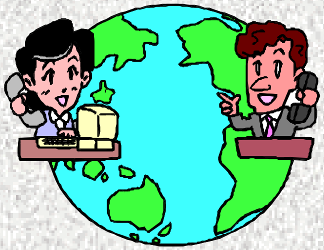
- Data types
    - integer (whole numbers only)
    - real (decimal)
    - character (alphabetic & numeric characters)
    - date
  - more advanced systems may include pictures & images as data types
    - Example: a database of buildings for the fire department which stores a picture as well as address,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



- Standard Operations
  - Examples: sort, delete, edit, select records

Database Management Systems: *Components* *(Continued)*

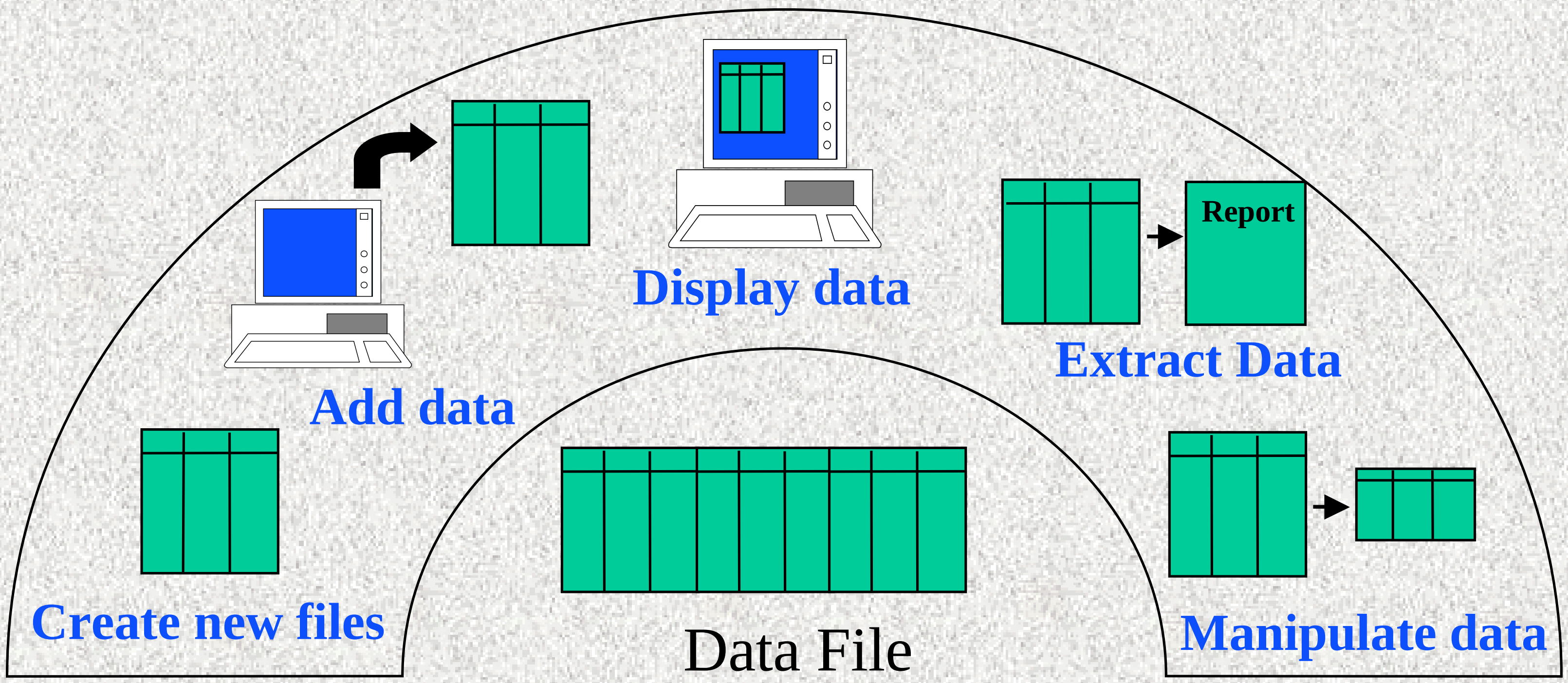
- Data definition Language (DDL)
  - The language used to describe the contents of the database
    - Examples: attribute names, data types - "Metadata"
- Data manipulation & Query Language
  - The language used to form commands for input, edit, analysis, output, reformatting, etc.
  - Some degree of standardization has been achieved with SQL (Standard Query Language)



Database Management Systems: *Components* *(Continued)*

- 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

Database Management System (DBMS) Functions



Database Management Systems
Types of Database Systems

- Several models for databases:
  - Tabular (“flat file”) - data in single table
  - Hierarchical
  - Network
  - Relational
- The hierarchical, network & 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

    - *Example:* database may need to handle information on aircraft, crew, flights, and passengers - four types of records with different attributes, but with relationships between them (“is booked on” between passenger & flight)

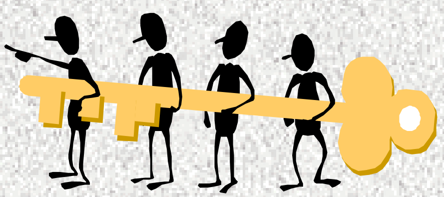
Relational Model

- 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
- Flexible approach to linkages between records comes closes to modeling the complexity of spatial relationships between objects
- Proposed by IBM researcher E.F. Codd (1970)
- More of a concept than a data structure
  - Internal architecture varies substantially from one RDBMS to another

Relational Model ~ 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
  - Each row is a record or tuple
  - Each column is an attribute

Relational Model ~ 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
  - Non-redundancy
    - No attribute in the key can be discarded without destroying the key's uniqueness
  - Example: phone number is a unique key in a phone directory
    - In a 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)

Types of Keys

- PRIMARY KEY
  - Serves as the sole row level addressing mechanism in the relational database model.
  - It can be formed through the combination of several items.
- FOREIGN KEY
  - A column or set of columns within a table that are required to match those of a primary key of a second table.
- Once linked, these keys form a RELATIONAL JOIN - thereby connecting row to row across the databases within which they reside.

Relational Database Management System (RDBMS)

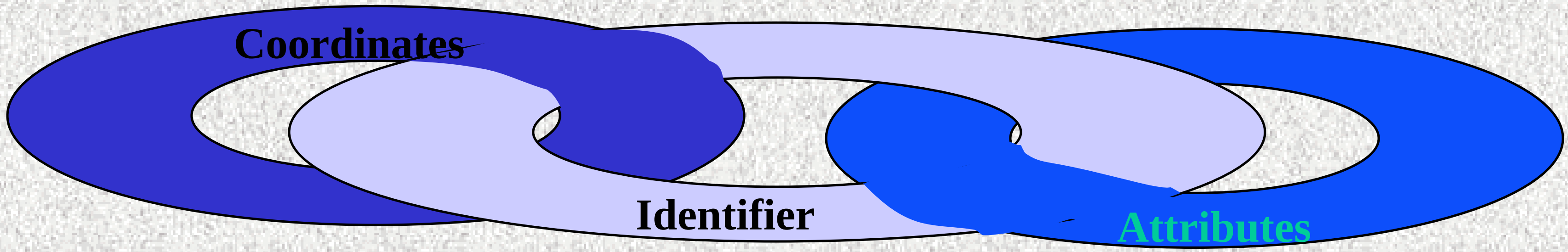
Table A

Name	Address	Parcel #
J. Smith	18 Lawers Dr.	216554
T. Brown	14 Seminole Tr.	887419

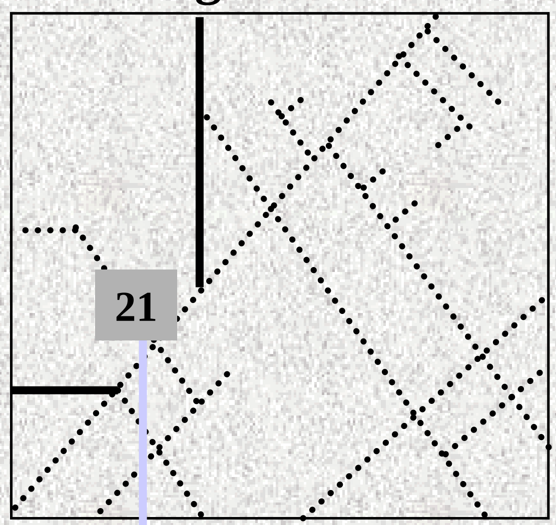
Table B

Parcel #	Assessed Value
887419	120,000
216397	100,000

Coverage Data Model



Coverage



Coordinates

IDENTIFIER	X,Y PAIRS
21	

Attributes

IDENTIFIER			
21			

Relating Data Files to the Feature Attribute Table

Soil.pat

AREA	PERIM	SO5AT_	SO5AT_ID
-10.0000	40.0000	1	0
5.0000	7.0000	2	298
2.5000	5.0000	3	300
2.5000	5.0000	4	301

Soil.data

SO5AT_ID	SOIL_TYPE
298	30
299	27
300	27
301	32
302	32

Joining Data Files
to the Feature Attribute Table

Relate Item

Start Item

linear | ordered | link

In_file

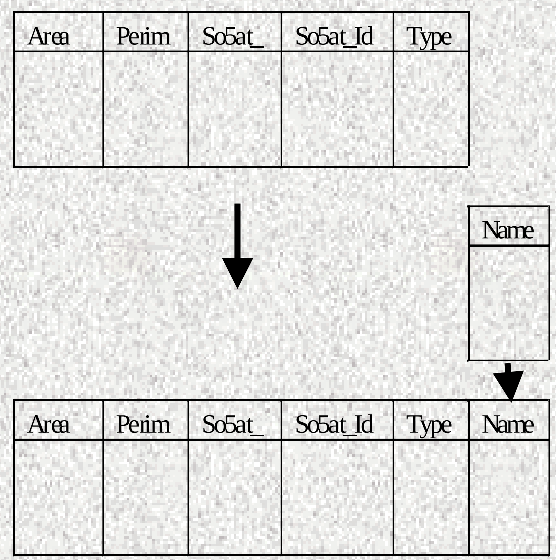
AREA	OERUN	SO5AT_	SO5AT_ID

Join_file

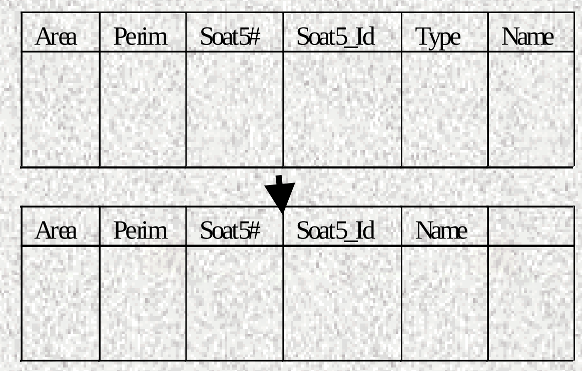
SO5AT_ID	TYPE	NAME	SUIT

AREA	PERIM	SO5AT_	SO5AT_ID	TYPE	NAME	SUIT

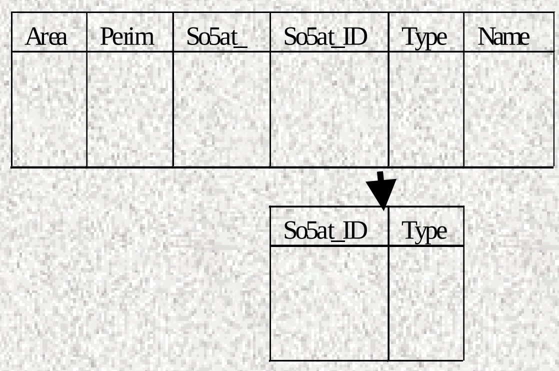
Manipulating the Data File Structure



ADDITEM



DROPITEM



PULLITEMS

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 the file, but allow access between files through common attributes

Example

- Given two tables:
  - 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

Standard Query Language (SQL)

- Different systems use different ways of formulating queries
- SQL is used by many systems
- SQL phrase structure:
  - SELECT, attribute name(s)>FROM<table> WHERE<condition statement>
    - Ex: SELECT FROM USE WHERE CLASS="U"
    - This selects only the objects for display - no attributes are retrieved by the query

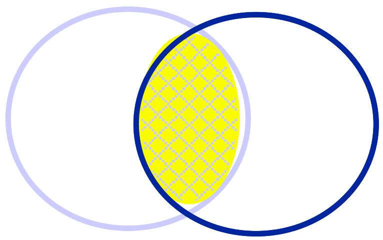
SQL (continued)

- SQL phrase structure (continued)
  - SQL examples using a list of student names
    - SELECT name FROM list (selects all names)
    - SELECT name FROM list WHERE grade = “A” (selects names of students receiving an “A”)
    - SELECT name FROM list WHERE cumgrade > 3.0 (selects names of students with a cumulative gpa greater than 3.0)
  - SQL operators:
    - Relational: <, >, =, <= ,>=
    - Arithmetic: =, -, *, / (only on numeric fields
    - Boolean: and, or, not

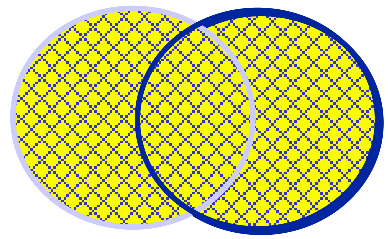
Boolean Operators

- Used to combine conditions
  - Ex: WHERE cumgrade > 3.0 AND grade = "A" (selects students satisfying both conditions only)
- Can have spatial meaning in GIS as well
  - Ex: when two maps are overlayed, areas (polygons) that are superimposed have the "and" condition
- A spatial representation is used to illustrate Boolean operators in the study of logic, through the use of diagrams called Venn diagrams
  - Thus GIS area overlay is a geographical instance of a Venn Diagram

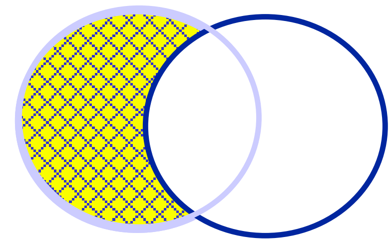
Boolean Operators



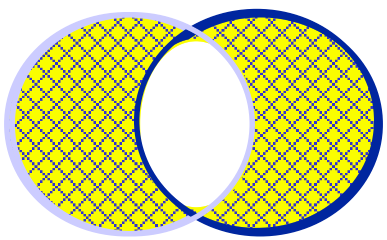
A AND B



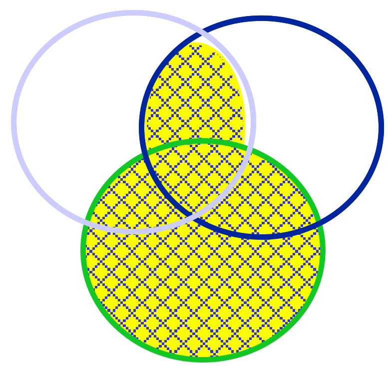
A OR B



A NOT B



A XOR B



(A AND B) OR C

SQL Extensions for Spatial Queries

- Some systems allow specifically spatial queries to be handled under SQL
  - Ex: WITHIN operator
    - SELECT <objects> WITHIN <specific area>
- The criteria for these spatial searches may include searching within the radius of a point, within a bounding rectangle, or within an irregular polygon