Chapter 9 – Geographic database

- 1 Introduction
- 2 Database management systems
- 3 Storing data in DBMS tables
- 4 SQL
- 5 Georaphic database types and functions
- 6 Geographic database design
- 7 Structuring geographic information
- 8 Editing and data maintenance
- 9 Conclusion

Learning Objectives

- After reading this chapter you will be able to:
  - 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

§ 1 Introduction

- A database is an integrated set of data on a particular subject
- Advantages over traditional file based datasets:
  - Assembling all data at a single location *reduces redundancy*
  - *Maintenance costs decrease* because of better organization and reduced data duplication
  - Applications become *data independent* so that multiple applications can use the same data and can evolve separately over time
  - *User knowledge* can be transferred between applications more easily because the database remains constant
  - *Data sharing* is facilitated and a corporate view of data can be provided to all managers and users
  - *Security and standards* for data and data access can be established and enforced
  - DBMS are better suited to managing large numbers of *concurrent users* working with vast amounts of data

§ 1 Introduction(cont.)

- Disadvantages to using databases when compared to files:
  - The cost of acquiring and maintaining DBMS software can be quite high
  - A DBMS adds complexity to the problem of managing data, especially in small projects
  - Single user performance will often be better for files, especially for more complex data types and structures where specialist indexes and access algorithms can be implemented
- This chapter describes how to create and maintain geographic databases, and the concepts, tools, and techniques that are available to manage geographic data in databases.
- Several other chapters provide additional information that is relevant to this discussion

§ 2 Database Management System(DBMS)

- A DBMS is a software application designed to organize the efficient and effective storage and access of data
- Database capabilities:
  - A Data model
  - A dara load capabilities
  - Indexes
  - A Query language
  - Security
  - Controlled update
  - Backup and recovery
  - Database administration tools
  - Application programming interfaces(APIs)

§ 2 Database Management System(cont.)

§ 2.1 Types of DBMS

- DBMS can be classified according to the way they store and manipulate data
- Three main types of DBMS are available to GIS users today
  - Relational (RDBMS)
  - Object(ODBMS)
  - Object-relational (ORDBMS)

§ 2 Database Management System(cont.)

§ 2.1 Types of DBMS(cont.)

- ORDBMS with geographic extensions:
- A query parser – is extended to deal with *geographic types and functions*
- A query optimizer – can handle geographic queries efficiently
- A query language – can handle geographic types (e.g., points and polygons) and functions (e.g., select polygons that touch each other)
- Indexing services – is extended to support multidimensional (i.e., *x*, *y*, *z* coordinates) geographic data types
- Storage management – the large volume of geographic records with different sizes (especially geometric and topological relationships) is accommodated through specialized storage structures
- Transaction services – standard DBMS are designed to handle short (sub-second) transactions and are extended to deal with the long transactions common in many geographic applications
- Replication –can deal with geographic types, and problems of reconciling changes made by distributed users

§ 2 Database Management System(cont.)

§ 2.2 Geographic DBMS extensions(DB side)

- Two vendors:
  - IBM: DB2 spatial extender, Informix Spatial Database
  - Oracle Spatial
- None of these is a complete GIS software system in itself
- The focus of extensions is data storage, retrieval, and management
- must be used in conjunction with a GIS except in the case of the simplest query-focused applications

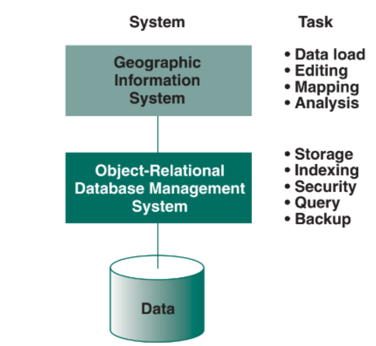
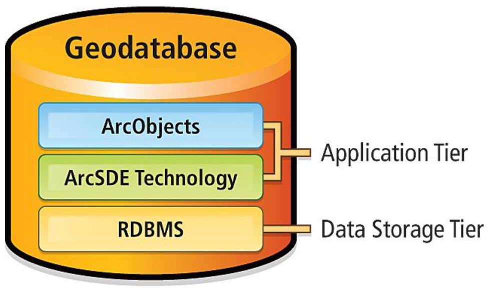


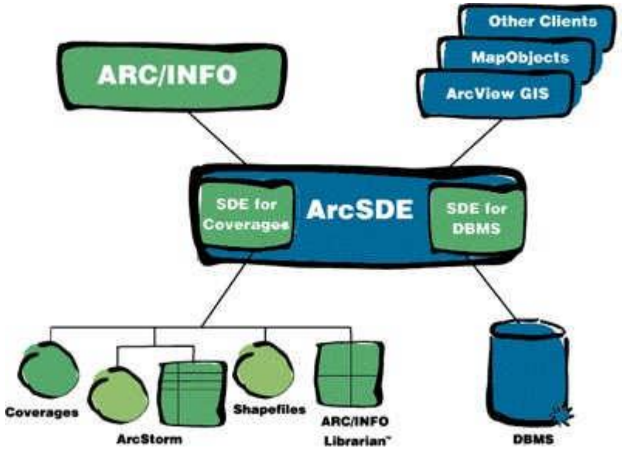
Figure 10.1 The roles of GIS and DBMS

§ 2 Database Management System(cont.)

§ 2.3 Geographic middleware extensions(GIS side)

- An alternative to extending the DBMS software kernel to manage geographic data is to build support for spatial data types and functions into a middle-tier (or middleware) application server
  - can also deliver better performance especially in the case of the more complex queries used in high-end GIS applications
  - both the DBMS and the application server hardware resources can be used in parallel

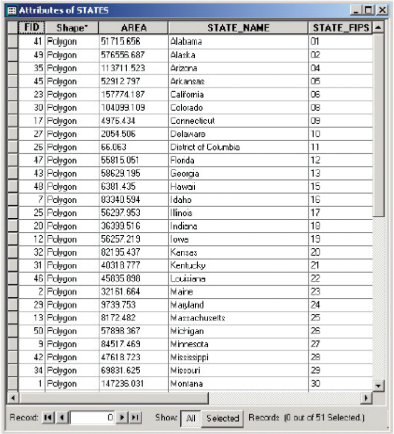




§ 3 Storing data in DBMS tables

- Relational databases are made up of tables. Each geographic class (layer) is stored as a table
- Database tables can be joined together to create new views of the database

(A)

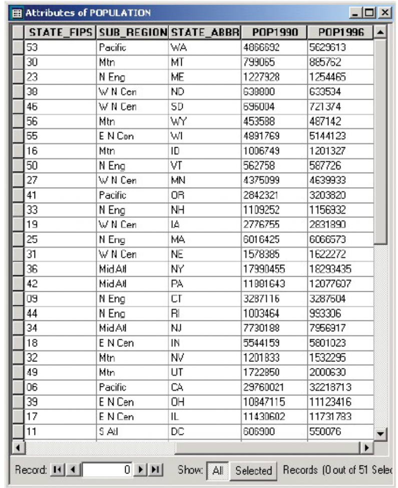


Attributes of STATES

FID	Shape*	AREA	STATE_NAME	STATE_FIPS
41	Polygon	51715.656	Alabama	01
49	Polygon	576556.687	Alaska	02
35	Polygon	113711.523	Arizona	04
45	Polygon	52912.797	Arkansas	05
23	Polygon	157774.187	California	06
30	Polygon	104099.109	Colorado	08
17	Polygon	4976.434	Connecticut	09
27	Polygon	2054.506	Delaware	10
26	Polygon	66.063	District of Columbia	11
47	Polygon	55815.051	Florida	12
43	Polygon	58629.195	Georgia	13
48	Polygon	6381.435	Hawaii	15
7	Polygon	83340.594	Idaho	16
25	Polygon	56297.953	Illinois	17
20	Polygon	36399.516	Indiana	18
12	Polygon	56257.219	Iowa	19
32	Polygon	82195.437	Kansas	20
31	Polygon	40318.777	Kentucky	21
46	Polygon	45835.898	Louisiana	22
2	Polygon	32161.664	Maine	23
29	Polygon	9739.753	Maryland	24
13	Polygon	8172.482	Massachusetts	25
50	Polygon	57898.367	Michigan	26
9	Polygon	84517.469	Minnesota	27
42	Polygon	47618.723	Mississippi	28
34	Polygon	69831.625	Missouri	29
1	Polygon	147236.031	Montana	30

Record: 0 Show: All Selected Records (0 out of 51 Selected.)

(B)

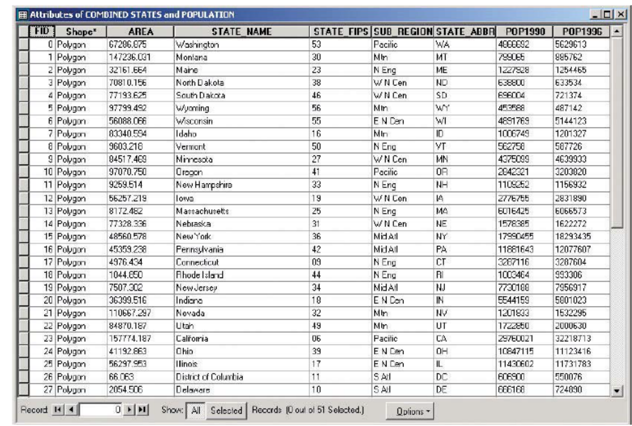


Attributes of POPULATION

STATE_FIPS	SUB_REGION	STATE_ABBR	POP1990	POP1996
53	Pacific	WA	4866692	5629613
30	Mtn	MT	799065	885762
23	N Eng	ME	1227928	1254465
38	W N Cen	ND	638800	633534
46	W N Cen	SD	696004	721374
56	Mtn	WY	453588	487142
55	E N Cen	WI	4891769	5144123
16	Mtn	ID	1006749	1201327
50	N Eng	VT	562758	587726
27	W N Cen	MN	4375099	4639933
41	Pacific	OR	2842321	3203820
33	N Eng	NH	1109252	1156932
19	W N Cen	IA	2776755	2831890
25	N Eng	MA	6016425	6066573
31	W N Cen	NE	1578385	1622272
36	Mid Atl	NY	17990455	18293435
42	Mid Atl	PA	11881643	12077607
09	N Eng	CT	3287116	3287604
44	N Eng	RI	1003464	993306
34	Mid Atl	NJ	7730188	7956917
18	E N Cen	IN	5544159	5801023
32	Mtn	NV	1201833	1532295
49	Mtn	UT	1722850	2000630
06	Pacific	CA	29760021	32218713
39	E N Cen	OH	10847115	11123416
17	E N Cen	IL	11430602	11731783
11	S Atl	DC	606900	550076

Record: 0 Show: All Selected Records (0 out of 51 Selec

(C)



Attributes of COMBINED STATES and POPULATION

FID	Shape*	AREA	STATE_NAME	STATE_FIPS	SUB_REGION	STATE_ABBR	POP1990	POP1996
0	Polygon	67286.875	Washington	53	Pacific	WA	4866692	5629613
1	Polygon	147236.031	Montana	30	Mtn	MT	799065	885762
2	Polygon	32161.664	Maine	23	N Eng	ME	1227928	1254465
3	Polygon	70810.156	North Dakota	38	W N Cen	ND	638800	633534
4	Polygon	77193.625	South Dakota	46	W N Cen	SD	696004	721374
5	Polygon	97799.492	Wyoming	56	Mtn	WY	453588	487142
6	Polygon	56088.066	Wisconsin	55	E N Cen	WI	4891769	5144123
7	Polygon	83340.594	Idaho	16	Mtn	ID	1006749	1201327
8	Polygon	9603.218	Vermont	50	N Eng	VT	562758	587726
9	Polygon	84517.469	Minnesota	27	W N Cen	MN	4375099	4639933
10	Polygon	97070.750	Oregon	41	Pacific	OR	2842321	3203820
11	Polygon	9259.514	New Hampshire	33	N Eng	NH	1109252	1156932
12	Polygon	56257.219	Iowa	19	W N Cen	IA	2776755	2831890
13	Polygon	8172.482	Massachusetts	25	N Eng	MA	6016425	6066573
14	Polygon	77328.336	Nebraska	31	W N Cen	NE	1578385	1622272
15	Polygon	48560.578	New York	36	Mid Atl	NY	17990455	18293435
16	Polygon	45359.238	Pennsylvania	42	Mid Atl	PA	11881643	12077607
17	Polygon	4976.434	Connecticut	09	N Eng	CT	3287116	3287604
18	Polygon	1044.650	Rhode Island	44	N Eng	RI	1003464	993306
19	Polygon	7507.302	New Jersey	34	Mid Atl	NJ	7730188	7956917
20	Polygon	36399.516	Indiana	18	E N Cen	IN	5544159	5801023
21	Polygon	110667.297	Nevada	32	Mtn	NV	1201833	1532295
22	Polygon	84870.187	Utah	49	Mtn	UT	1722850	2000630
23	Polygon	157774.187	California	06	Pacific	CA	29760021	32218713
24	Polygon	41192.863	Ohio	39	E N Cen	OH	10847115	11123416
25	Polygon	56297.953	Illinois	17	E N Cen	IL	11430602	11731783
26	Polygon	66.063	District of Columbia	11	S Atl	DC	606900	550076
27	Polygon	2054.506	Delaware	10	S Atl	DE	666168	724890

Record: 0 Show: All Selected Records (0 out of 51 Selected.) Options

Figure 10.2 *(continued)*

Figure 10.2 GIS database tables for US States: (A) STATES table; (B) POPULATION table; (C) joined table – COMBINED STATES and POPULATION

§ 3 Storing data in DBMS tables(cont.)

- From conceptual to logical : normalizations

(A)

ParcelNumb	OwnerNam	OwnerAddress	PostalCode	ZoningCode	ZoningType	Date / AssessedValue
673/100	Jeff Peters	10 Railway Cuttings	114390	2	Residential	2002 220000
673-101	Joel Campbell	1115 Center Place	114390	2	Residential	2003 545500
674-100	Dave Widseler		114391	3	Commercial	99 249000
674-100		452 Diamond Plaza	114391	3	Commercial	2000 275500
674-100	D Widseler	452 Diamond Plaza	114391	3	Commercial	2001 290000
670-231	Sam Camarata	19 Big Bend Bld	114391	2	Residential	2004 450575
674-112	Chris Capelli	Hastings Barracks	114392	2	Residential	2004 350000
674-113	Sheila Sullivan	10034 Endin Mansions	114391	2	Residential	02 1005425

Figure 10.3 Tax assessment database: (A) raw data; (B) cleansed data in a GIS DBMS; (C) data partially normalized into three sub-tables; (D) joined table

(B)

Attributes of Tab10_2

OBJECTID*	ParcelNumb	OwnerNam	OwnerAddress	PostalCode	ZoningCode	ZoningType	DateAssessed	AssessedValue
1	673-100	Jeff Peters	10 Railway Cuttings	114390	2	Residential	2002	220000
2	673-101	Joel Campbell	1115 Center Place	114390	2	Residential	2003	545500
3	674-100	Dave Widseler	452 Diamond Plaza	114391	3	Commercial	1999	249000
4	674-100	Dave Widseler	452 Diamond Plaza	114391	3	Commercial	2000	275500
5	674-100	Dave Widseler	452 Diamond Plaza	114391	3	Commercial	2001	290000
6	670-231	Sam Camarata	19 Big Bend Bld	114391	2	Residential	2004	450575
7	674-112	Chris Capelli	Hastings Barracks	114392	2	Residential	2004	350000
8	674-113	Sheila Sullivan	10034 Endin Mansions	114391	2	Residential	2002	1005425

Record: 0 Show: All Selected Records (0 out of 8 Selected) Options

§ 3 Storing data in DBMS tables(cont.)

- Joining of tables

(C)

Attributes of Tab10_3a

OBJECTID*	ParcelNumb	ZoningCode	DateAssessed	AssessedValue	OwnersName
1	673-100	2	2002	222000	Jeff Peters
2	673-101	2	2003	545500	Joel Campbell
3	674-100	3	1999	249000	Dave Widseler
4	674-100	3	2000	275500	Dave Widseler
5	674-100	3	2001	290000	Dave Widseler
6	670-231	2	2004	450575	Sam Camarata
7	674-112	2	2004	350000	Chris Capelli
8	674-113	2	2002	1005425	Sheila Sullivan

Record 8 Show: All Selected Records (0 out of 8 Selected) Options

Attributes of Tab10_3b

OBJECTID*	ZoningCode	ZoningType
1	2	Residential
2	3	Commercial

Record: 1 Show: All Selected Records (0 out of 2 Selected.)

Attributes of Tab10_3c

OBJECTID*	OwnerName*	Address	PostalCode
2	Jeff Peters	10 Railway Cuttings	114390
3	Joel Campbell	1115 Center Place	114390
4	Dave Widseler	452 Diamond Plaza	114391
5	Sam Camarata	19 Big Bend Bld	114391
6	Chris Capelli	Hastings Barracks	114392
7	Sheila Sullivan	10034 Endin Mansions	114391

Record: 0 Show: All Selected Records (0 out of 6 Selected.)

(D)

Attributes of Tab10_3a_Tab10_3b_Tab10_3c

OBJECTID	ParcelNumb	DateAssessed	AssessedValue	ZoningType	ZoingCode	OwnerName	OwnerAddress	PostalCode
1	673-100	2002	222000	Residential	2	Jeff Peters	10 Railway Cuttings	114390
2	673-101	2003	545500	Residential	2	Joel Campbell	1115 Center Place	114390
3	674-100	1999	249000	Commercial	3	Dave Widseler	452 Diamond Plaza	114391
4	674-100	2000	275500	Commercial	3	Dave Widseler	452 Diamond Plaza	114391
5	674-102	2004	290000	Residential	2	Dave Widseler	452 Diamond Plaza	114391
6	670-231	2004	450575	Residential	2	Sam Camarata	19 Big Bend Bld	114391
7	674-112	2004	350000	Residential	2	Chris Capelli	Hastings Barracks	114392
8	674-113	2002	1005425	Residential	2	Sheila Sullivan	10034 Endin Mansions	114391

Record: 0 Show: All Selected Records (0 out of 8 Selected) Options

Figure 10.3 *(continued)*

§ 4 SQL

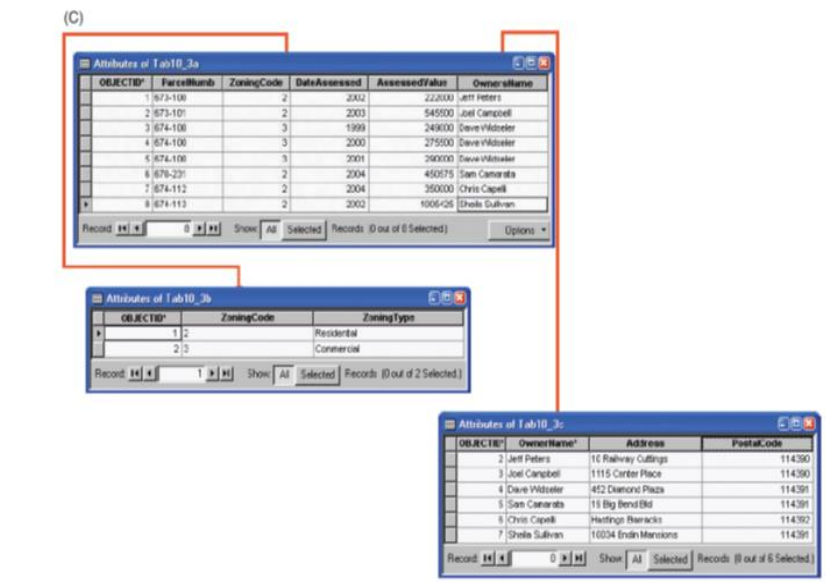
- SQL is the standard database query language

```
SELECT Tab10_3a.ParcelNumb, Tab10_3c.Address,
   Tab10_3a.AssessedValue
FROM (Tab10_3b INNER JOIN Tab10_3a ON
   Tab10_3b.ZoningCode =
   Tab10_3a.ZoningCode) INNER JOIN Tab10_3c
   ON Tab10_3a.OwnersName =
   Tab10_3c.OwnerName
WHERE (((Tab10_3a.AssessedValue)>300000) AND
   ((Tab10_3b.ZoningType)="Residential"));
```

```
CREATE TABLE Countries (
name          VARCHAR(200) NOT NULL PRIMARY
   KEY,
shape         POLYGON NOT NULL
CONSTRAINT  spatial reference
CHECK       (SpatialReference(shape) = 14)
)
```

```
INSERT INTO Countries
(Name, Shape) VALUES ('Kenya', Polygon('((x
   y, x y, x y, x y)) ,2))
```

```
SELECT Countries.Name,
FROM Countries
WHERE Area(Countries.Shape) > 11000
```



§ 5 Geographic database types and functions

- There have been several attempts to define a superset of geographic *data types* (variation of *point, line* and *area*)that can represent and process geographic data in databases
- Unfortunately space does not permit a review of them all
- Model developed International Standards Organization (ISO) and the Open Geospatial Consortium(OGC) standards

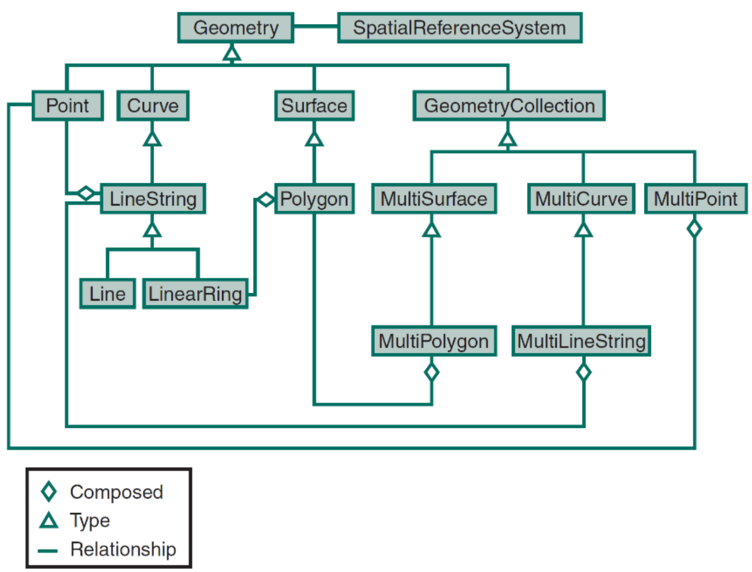
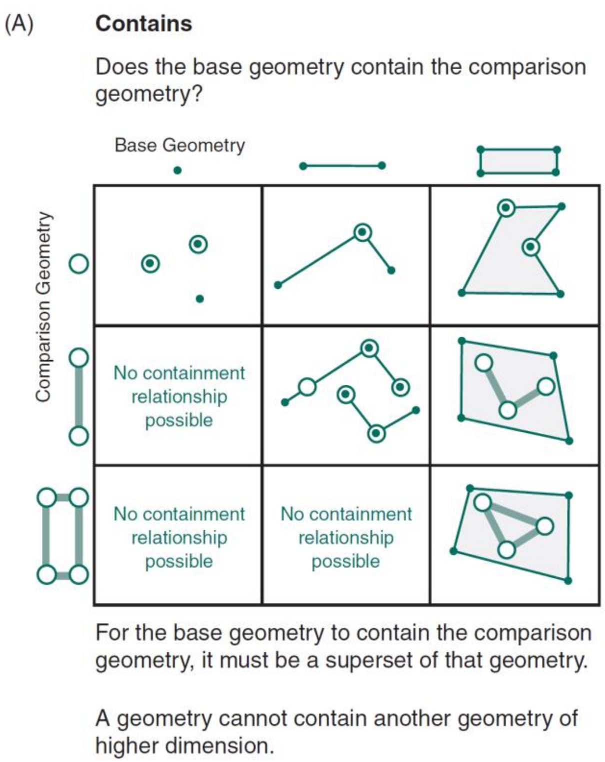


Figure 10.5 Geometry class hierarchy (*Source*: after OGC 1999) (Reproduced by permission of Open Geospatial Consortium, Inc.)

§ 5 Geographic database types and functions(cont.)

- Nine *functions* for testing relationship
- Equals, Disjoint, Intersects, Touches, Crosses,Within, Constains, Overlaps, Relate



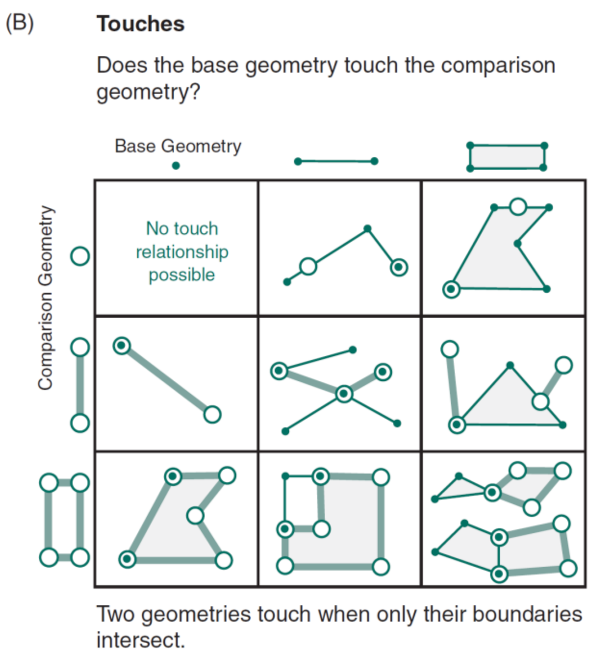
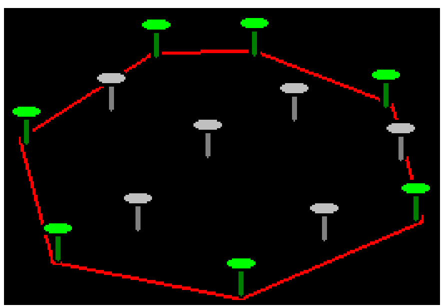
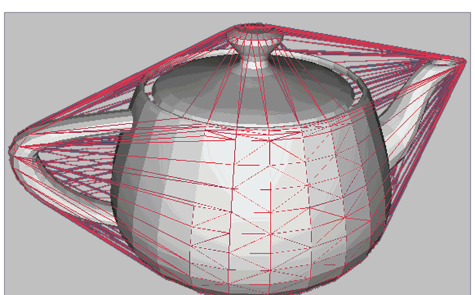


Figure 10.6 Examples of possible relations for two geographic database operators: (A) Contains; and (B) Touches operators (*Source*: after Zeiler 1999)

§ 5 Geographic database types and functions(cont.)

- Seven *functions* for geometric analysis
- Distance, Buffer, Convex Hull, Intersection, Union, Difference, SymDifference





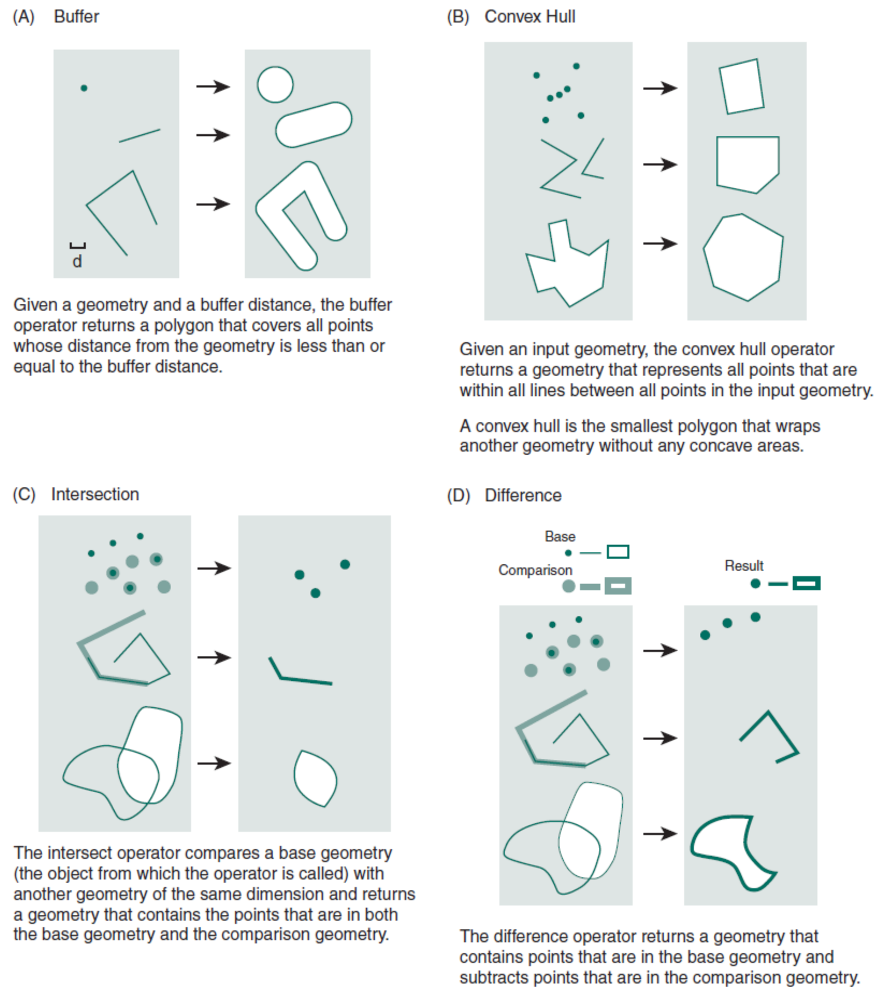


Figure 10.7 Examples of spatial analysis methods on geometries: (A) Buffer; (B) Convex Hull; (C) Intersection; (D) Difference (*Source*: after Zeiler 1999)

§ 6 Geographic database design

- 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*

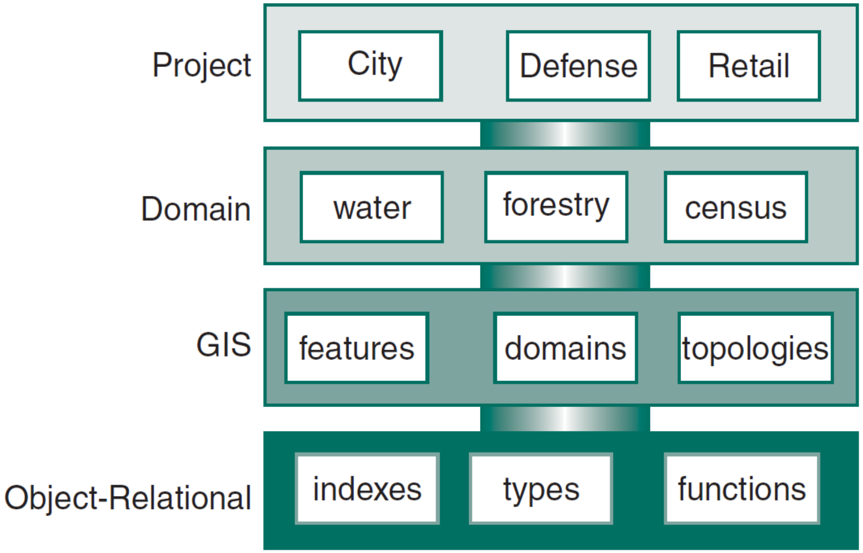


Figure 10.8 Four levels of data model available for use in GIS projects with examples of constructs used

§ 6 Geographic database design(cont.)

- Database design involves three key stages
  - conceptual, logical, and physical

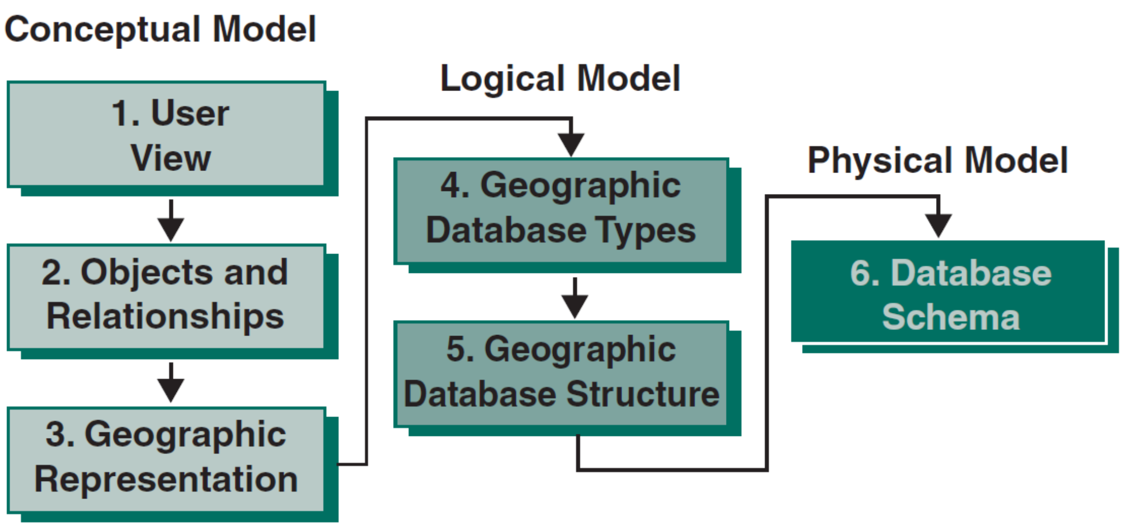


Figure 10.9 Stages in database design (*Source*: after Zeiler 1999)

§ 7 Structuring geographic information

§ 7.1 Topology creation

- Topology can be created for vector datasets using either batch or interactive techniques
  - Batch topology builders are required to handle CAD, survey, simple feature, and other unstructured vector data imported from non-topological systems, an iterative process because manual editing is required to make corrections
  - Interactive topology creation is performed dynamically at the time objects are added to a database using GIS editing software
- Two database-oriented approaches have emerged in recent years for storing and managing topology: Normalized and Physical

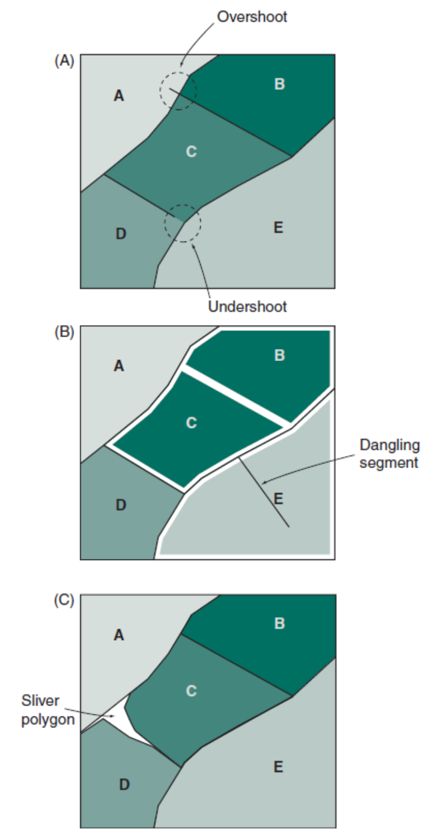
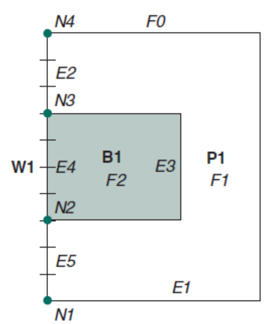


Figure 9.9 Examples of human errors in digitizing: (A) undershoots and overshoots; (B) invalid polygons; and (C) sliver polygons

§ 7 Structuring geographic information (cont.)

§ 7.1.1 Normalized model(topological)

- focuses on the storage of an arc-node data structure
  - 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
  - e.g., polygon objects are assembled at query time by joining together tables containing the line segment geometries and topological primitives that define topological relationships



Parcels

ID
P1

Buildings

ID
B1

Walls

ID
W1

Parcel x Face

Parcel	Face
P1	F1
P1	F2

Wall x Edge

Wall	Edge	Order	Orientation
W1	E2	1	+
W1	E4	2	-
W1	E5	3	+

Building x Face

Parcel	Face
B1	F2

Nodes

ID
N1
N2
N3
N4

Edges

ID	Vertices	From	To	Left	Right
E1	(0,10),(8,10),(8,0),(0,0)	N4	N1	F0	F1
E2	(0,10),(0,7)	N4	N3	F1	F0
E3	(0,7),(5,7),(5,3),(0,3)	N3	N2	F1	F2
E4	(0,3),(0,7)	N2	N3	F0	F2
E5	(0,3),(0,0)	N2	N1	F1	F0

Faces

ID
F0
F1
F2

Figure 10.10 Normalized database topology model

§ 7 Structuring geographic information (cont.)

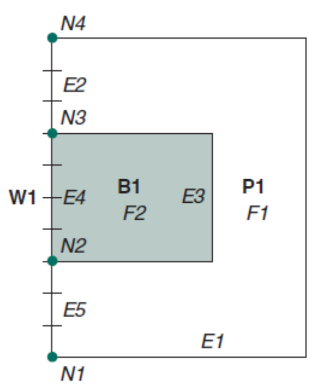
§ 7.1.1 Normalized model(cont.)

- Advantages
  - Similar to topological structure supporting spatial relationship
  - Easy to understand
  - Storage efficient
  - easily lend itself to access via a SQL API
- Disadvantages
  - Query performance
  - Management of topological structure in DBMS
  - Update issues
- Implemented in Oracle Spatial and can be accessed via a SQL API making it easily available to a wide variety of users

§ 7 Structuring geographic information (cont.)

§ 7.1.2 Physical model(spaghette)

- Topological primitives are not stored in the database
  - Topological relationships are computed on-the-fly whenever they are required by client applications



Parcels

ID	Vertices
P1	(0,0),(0,3),(0,7),(0,10),(8,10),(8,0),(0,0)

Buildings

ID	Vertices
B1	(0,3),(0,7),(5,7),(5,3),(0,3)

Walls

ID	Vertices
W1	(0,10),(0,7),(0,3),(0,0)

Topology Rules

ID	Rule Type
R1	Parcels no overlap
R2	Buildings no overlap

Topology Errors

ID	Vertices	Rule Type	Feature	IsException
E1	…	-	-	-

Dirty Area

Vertices
…

Figure 10.11 Physical database topology model

§ 7 Structuring geographic information (cont.)

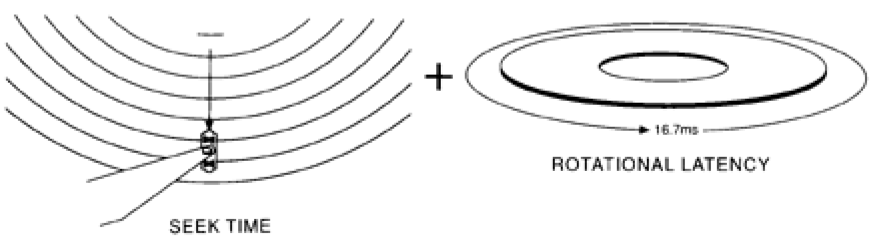
§ 7.1.2 Physical model(cont.)

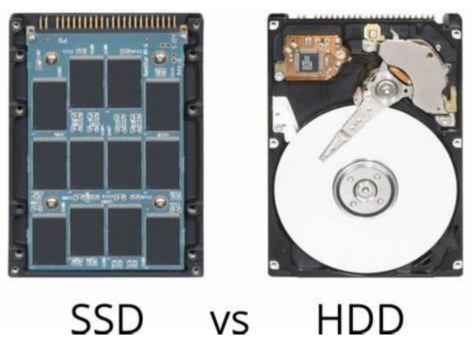
- Topological primitives are not stored in the database
  - Physical model requires that an external client or middle-tier application server is responsible for validating the topological integrity of datasets
  - Topologically correct features are then stored in the database using a much simpler structure than the Normalized model
  - When compared to the Normalized model, the Physical model offers two main advantages of simplicity and performance
  - Even when topology is required it is faster to retrieve feature geometries and re-compute topology outside the database
- Implemented in ESRI ArcGIS and offers fast update and query performance for high-end GIS applications
- ESRI has also implemented a long transaction and versioning model based on the physical database topology model

§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods

- A typical database consists of a collection of files, which contain a set of records, which are stored on a magnetic disk
- A disk block is the atomic unit of storage
- The transfer time of a disk block to or from a disk has three measurement components
  - Seek time - time to move disk heads to the correct track
  - Latency - rotation time of the disk to the correct position
  - CPU transfer time - time for the block transfer to the CPU

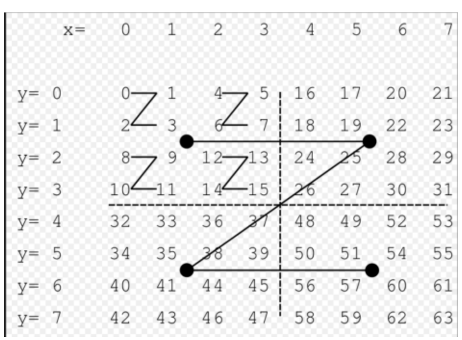




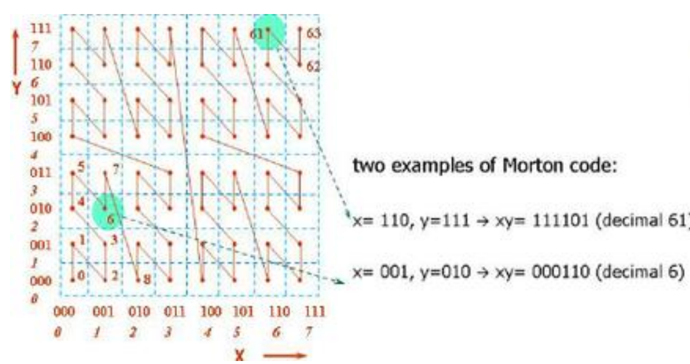
§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods（cont.）

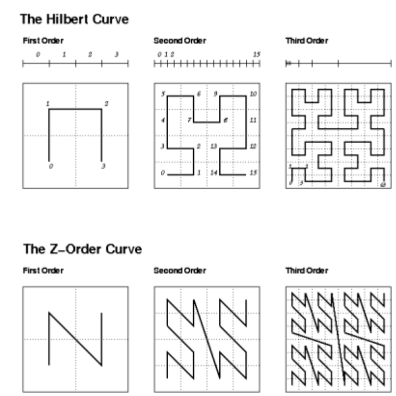
- Seek time being the dominant factor, performance may be maximized when the mechanical movement of disk heads is kept to a minimum
  - Data to be accessed together should be physically stored close together
  - Proper indexing can focus the search



Z-order Curve



Morton code



Hilbert Curve

§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods（cont.）

- Records, fields, file organization and access methods
  - Record - a collection of data items describing some logical entity
  - Field - a place for a data item
    - Variable-length
    - Fixed-length
    - Keys
  - File - a collection of records usually of the same type
    - Fixed-length or variable-length

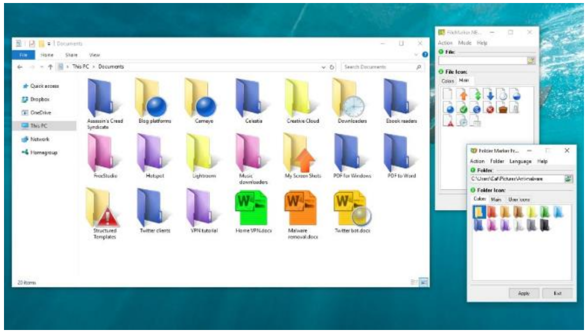


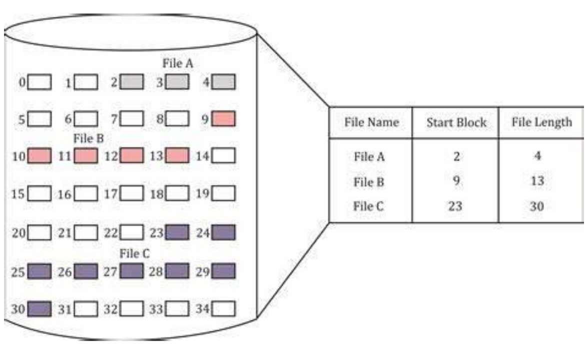
Table1

ID	FirstName	Surname	Age
1	John	Jones	35
2	Tracey	Smith	25
3	Anne	McNeil	30
4	Andrew	Francis	37
5	Gillian	Carpenter	32
6	Karen	Rogers	22
7	Amy	Sanders	42
8	Kevin	White	38
9	Charlie	Anderson	40
10	Mary	Brown	26
11	Andrew	Smith	32
12	James	Francis	28
13	Karen	Jones	30
14	Edward	Kent	32
15	Jenny	Smith	26
16	Angela	Jones	41
(New)			

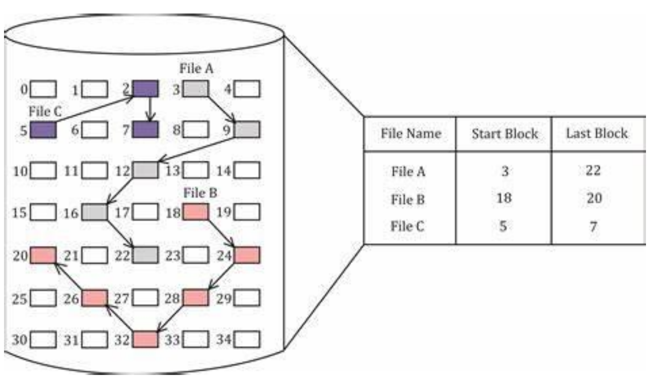
§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods（cont.）

- Files are stored on disk blocks
  - Contiguous allocation
  - Linked allocation
- File organization describes the physical organization of records on the disk and the way that the blocks of records are linked
- Access methods provide the means for application programs to manipulate files



Contiguous allocation



Linked allocation

§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods（cont.）

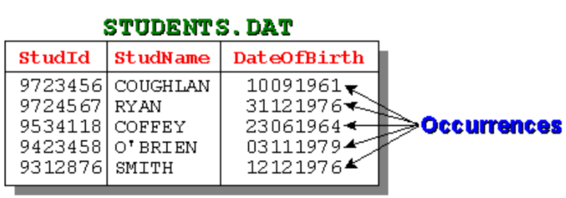
- Access methods
  - Open - open a file
  - Find - finds the block with the required record and puts it in main memory
  - Read - reads the record from main memory into the program memory
  - Delete - deletes records in memory and writes updated block to disk
  - Modify - updates a field in a record and writes back to disk
  - Insert - inserts a record and writes back to disk
  - Close - close a file

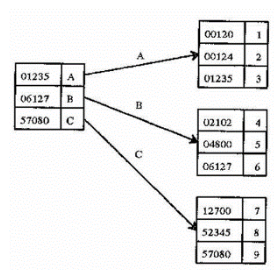
```
typedef struct _구조체
{
    int     a;
    short   b;
    float   f;

}구조체;

void LoadFP(const char* _pszName)
{
    FILE* fp;
    char*   pszName = "불러드릴파일경로";
    fp = fopen(pszName, "rb");

    int iTest1;
    int iRead;
    iRead = fread(&iTest1, 4, 1, fp);

    fseek(fp, 0, SEEK_SET);

    구조체 StructTemp;
    iRead = fread(&iTest1, 4, 1, fp);
    iRead = fread(&StructTemp, sizeof(StructTemp), 1, fp);

    fclose( fp );
}
```

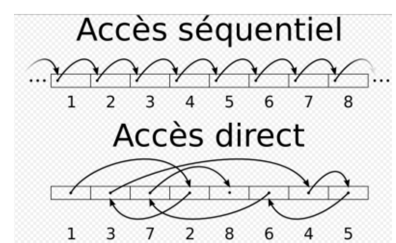
§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods（cont.）

- Four basic types of file organization
  - Unordered file
    - New records are inserted in the next physical position on the disk
    - Insertions are efficient but subsequent retrievals must search through the data sequentially
  - Ordered (sequential) file
    - New records are inserted according to the values in one or more fields - e.g., sorted on name or rank
    - Ordered files allow efficient binary searches
    - New insertions to sequential files are expensive
  - Random file
    - A file organized via an index. Also called a "direct file" or a "direct access file," it enables qui ck access to specific records within the file
    - The index points to a specific location, and the file is read from that point



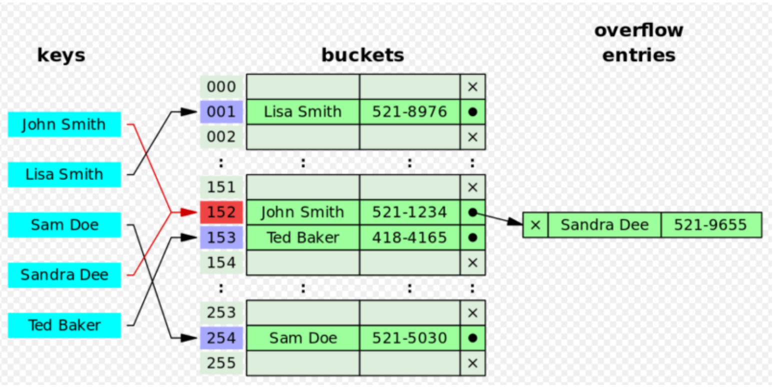


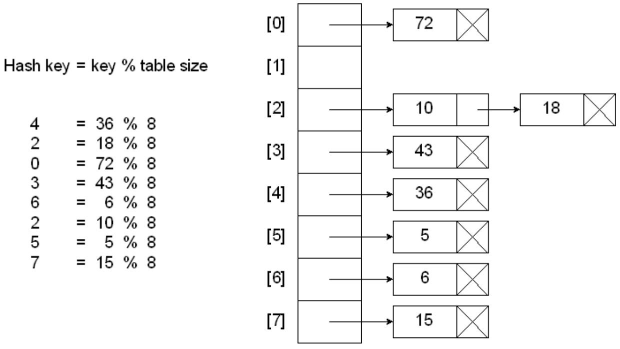


§ 7 Structuring geographic information (cont.)

§ 7.2 General Structures and Access Methods（cont.）

- Three basic types of file organization
  - Hashed files
    - Records are inserted at a disk location according to an address derived by a hash function
    - Retrievals repeat the hash function to find the disk location which requires only one disk access
    - One common method of determining a hash key is the **division method** of hashing(e.g., 质数除余法)



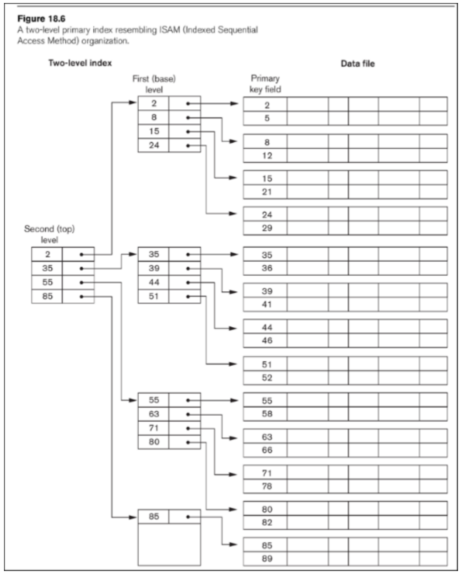


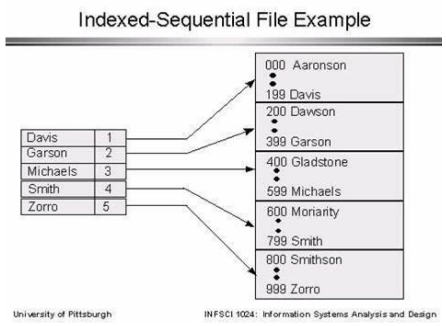
Hashed files

§ 7 Structuring geographic information (cont.)

§ 7.3 Indexing

- A special representation of information about objects that improves searching
- Similar in concept to the index of a book
- The index acts on one or more fields of the data file called the indexing field
- A single-level index is an ordered file with each record containing
  - Index field - the value of the indexing field
  - Pointer field - the address where the data lie
- Indexing allows binary search and logarithmic search times





§ 7 Structuring geographic information (cont.)

§ 7.3 Indexing(cont.)

- standard index in DBMS is about how key is related to the address of the record

  Key(s) = f (address)

- not suitable for GIS
- Spatial index is about how key is related to the place, where place is usually a component of space partition(grid, box) to which keys are assigned

  Key(s) = f (place)

- Because keys assigned to the grid are subset of all objects in the database, speed to find the right object among the subset is much faster
- Four main methods of general practical importance have emerged in GIS: MBR, grid indexes, quadtrees, and R-trees

§ 7 Structuring geographic information (cont.)

§ 7.3.1 Minimum Bounding Rectangle(MBR)

- Partitions the whole space into MBR related to each object
- Efficient means to approximate spatial objects and streamline searches
- Detailed geometry stored separately

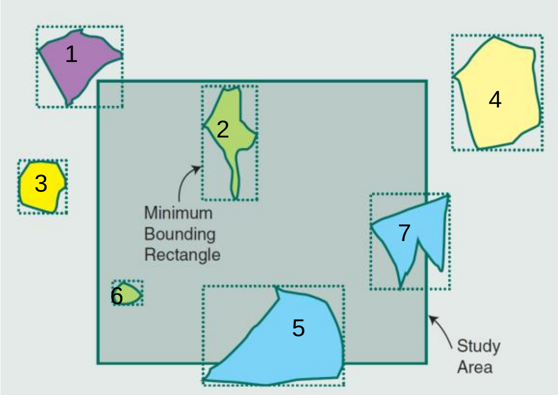


Figure 10.18 Polygon in polygon test using MBR. A MBR can be used to determine objects definitely within the study area (green) because of no overlap, definitely out (yellow), or possibly in (blue). Objects possibly in can then be analyzed further using their exact geometries. Note the purple object that is actually completely outside, although the MBR suggests it is partially within the study area

MBR ID	Coordinates	Object ID
1	$x_{1min},y_{1min},x_{1max},y_{1max}$	1
2	$x_{2min},y_{2min},x_{2max},y_{2max}$	2
3	$x_{3min},y_{3min},x_{3max},y_{3max}$	3
4	$x_{4min},y_{4min},x_{4max},y_{4max}$	4
5	$x_{5min},y_{5min},x_{5max},y_{5max}$	5
6	$x_{6min},y_{6min},x_{6max},y_{6max}$	6
7	$x_{7min},y_{7min},x_{7max},y_{7max}$	7

§ 7 Structuring geographic information (cont.)

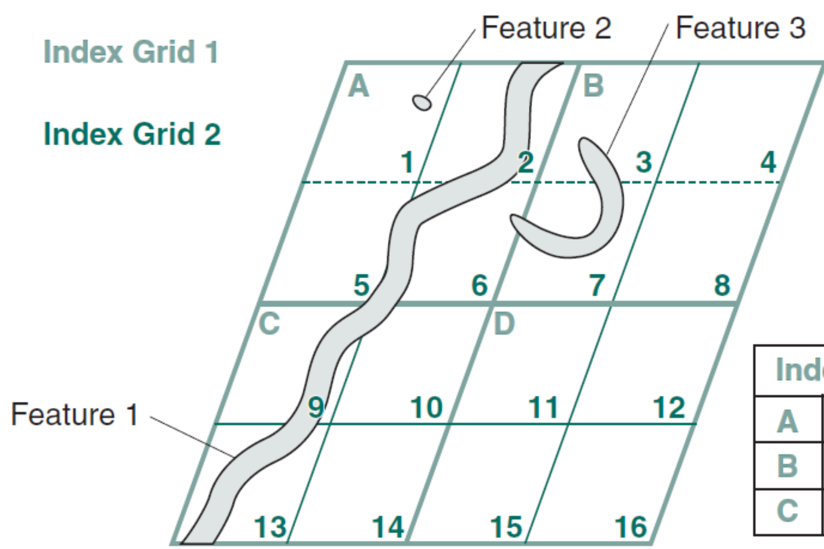
§ 7.3.2 Grid indexes

- A partition of the plane into equal sized cells
- Objects that fall within a particular cell are stored in a contiguous location
- Cell size determination
  - Number of observations is related to the number of cells; cells have a finite associated storage space
  - Types of expected queries: cell size should support the expected precision of query ranges
- An effective structure if there is an even spatial distribution of data; otherwise there are numerous empty cells and corresponding unused disk space

§ 7 Structuring geographic information (cont.)

§ 7.3.2 Grid indexes(cont.)

- Grid indexes are easy to create, can deal with a wide range of object types, and offer good performance.



Index 1	
A	1,2,3
B	3
C	1

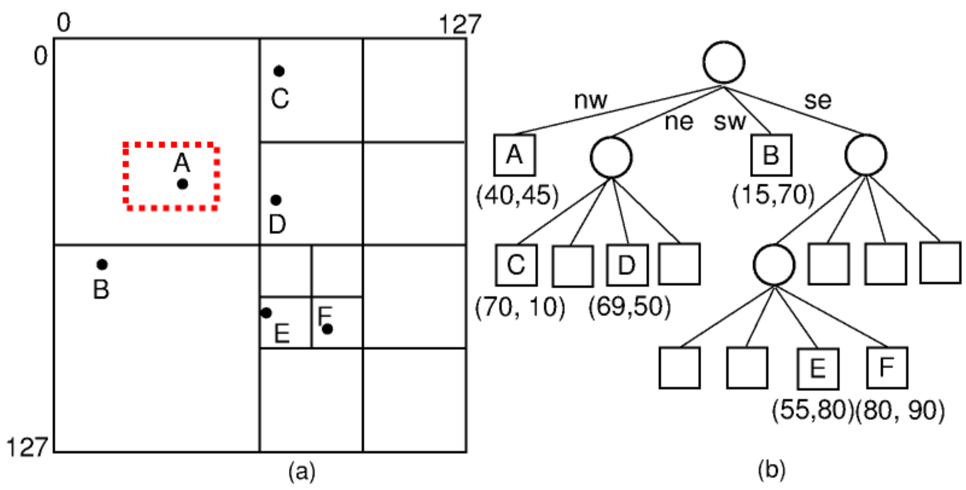
Index 2	
1	2
2	1
3	3
5	1
6	1,3
7	3
9	1
10	1
13	1

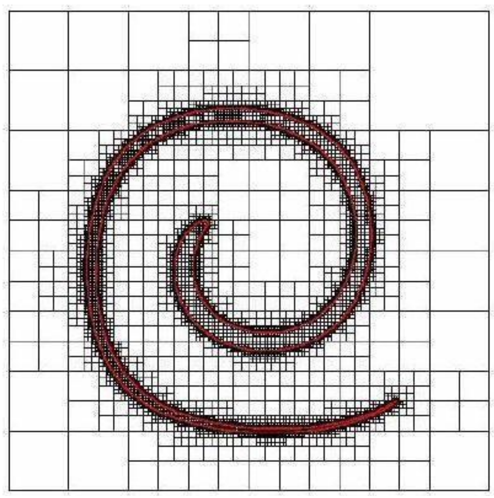
Figure 10.13 A multi-level grid geographic database index

§ 7 Structuring geographic information (cont.)

§ 7.3.3 Quadtree indexes

- In many respects quadtrees are a special types of grid
- So quadtree indexes could be regarded as the compression of grid indexes
- Save space and adaptive to the distribution of geo-objects

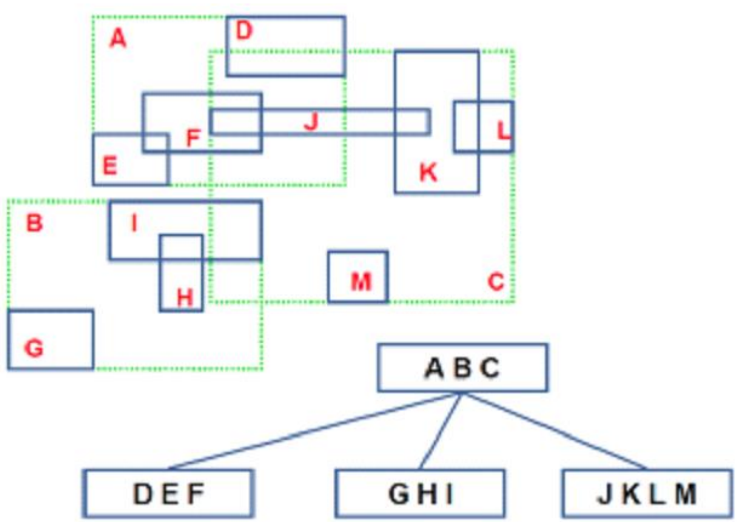


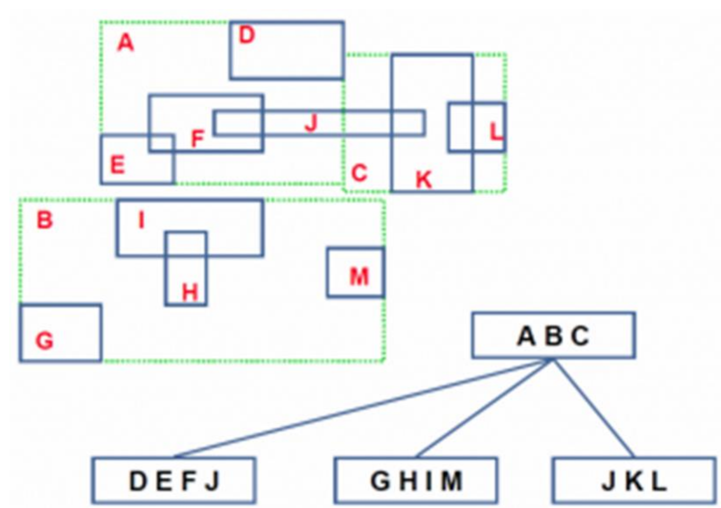


§ 7 Structuring geographic information (cont.)

§ 7.3.4 R-tree and R+-tree indexes

- Each node in the tree represents a rectangle
- Non-leaf nodes represent R-tree rectangles that contain the rectangles of its descendants
- Leaf nodes represent the actual rectangles to be indexed
- R+-trees do not permit overlapping of containing rectangles

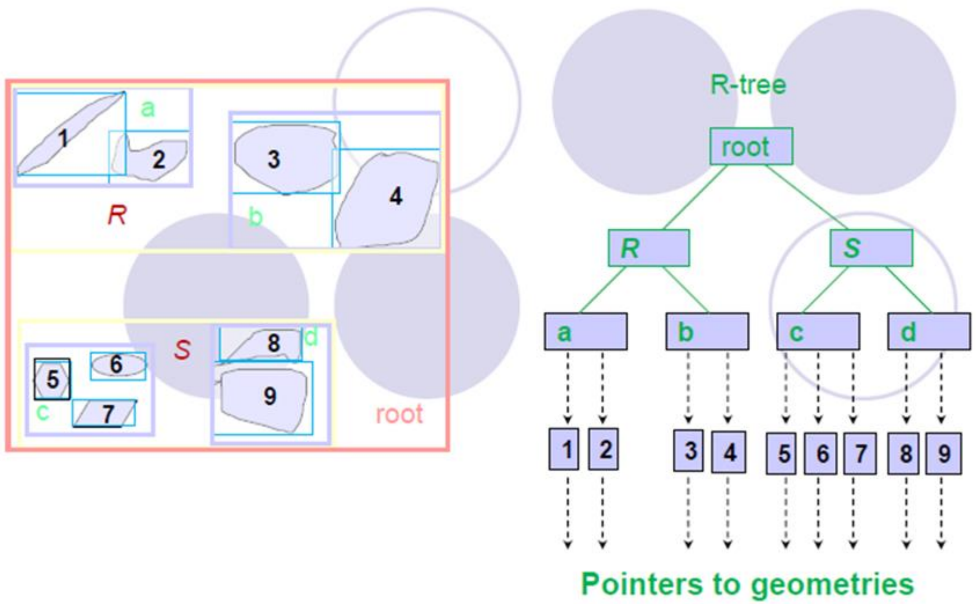




§ 7 Structuring geographic information (cont.)

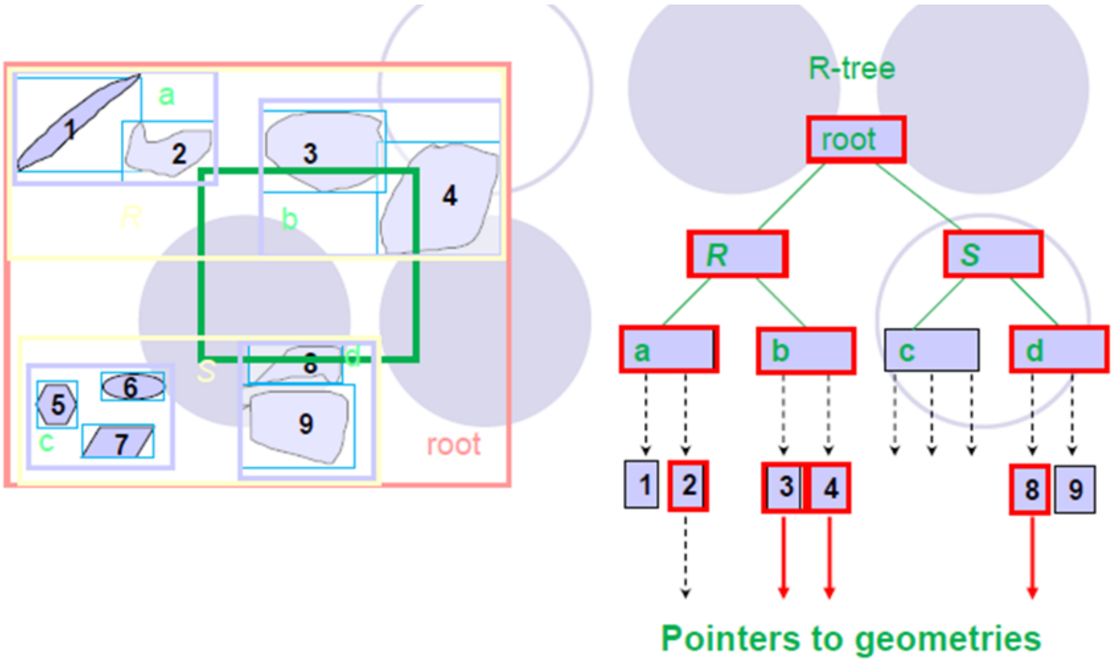
§ 7.3.4 R-tree and R^+-tree indexes(cont.)

- Based on MBRs
- MBRs of geometric objects form the leaves of the index
- Multiple MBRs are grouped into larger rectangles(MBRs) to form intermediate nodes in the index tree
- Repeat until one rectangle is left that contains everything



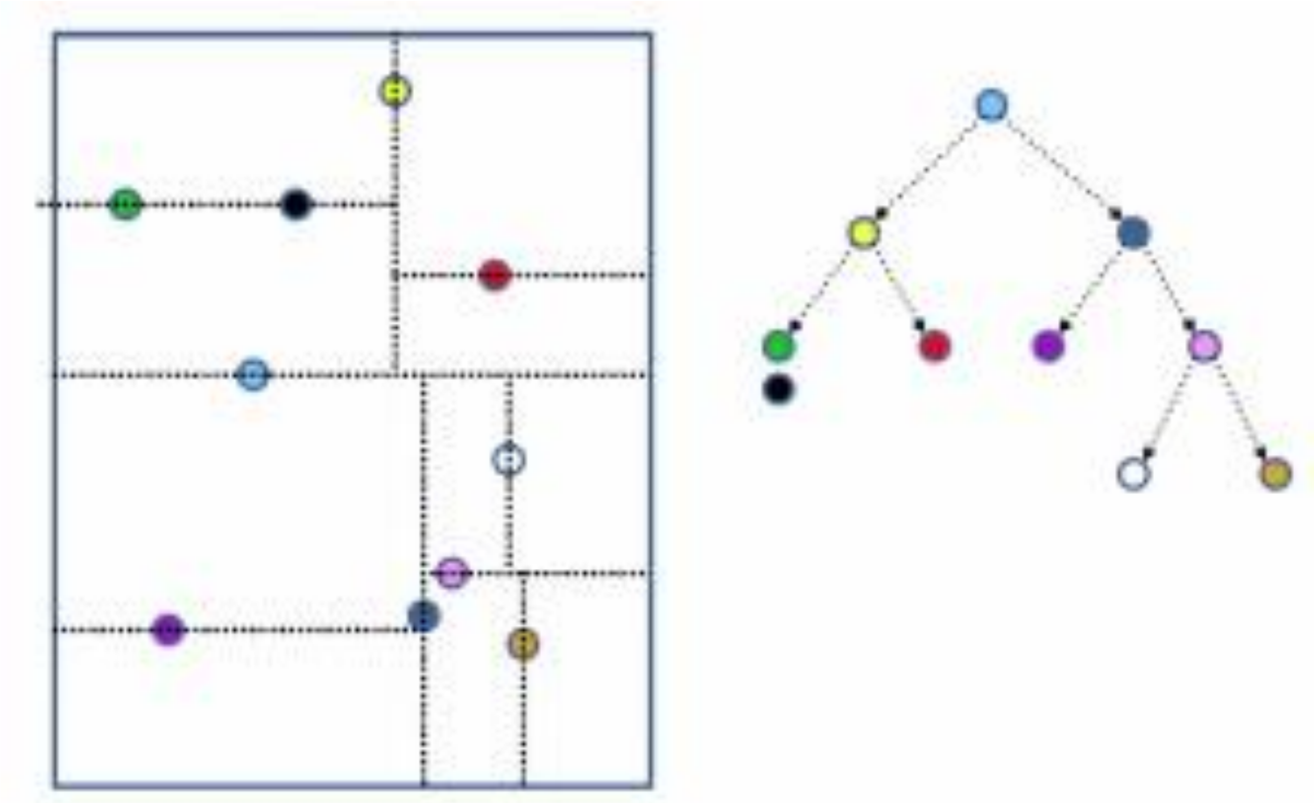
§ 7.3.4 R-tree and R^+-tree indexes(cont.)

- Consider two types of querie：
- Point query: what object contains the query point?
- Window query: what objects intersects the query window?



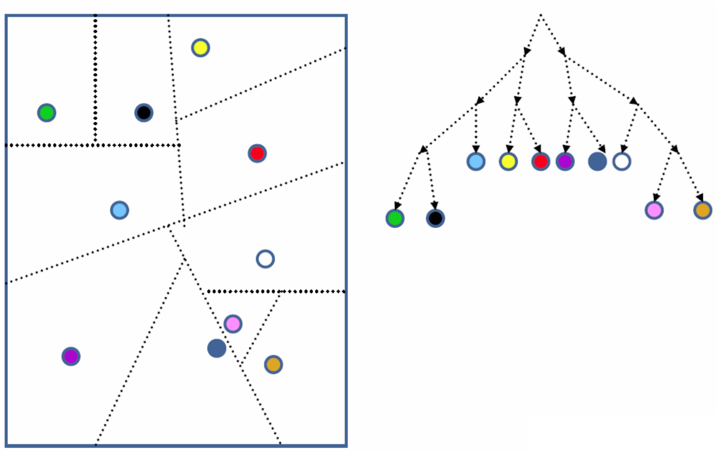
§ 7 Structuring geographic information (cont.)

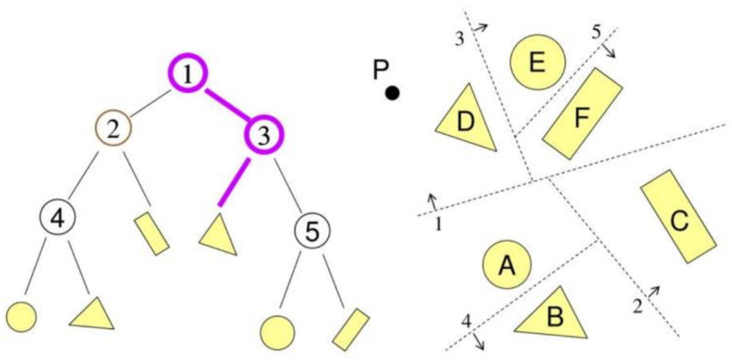
Other Index: K-D Tree



§ 7 Structuring geographic information (cont.)

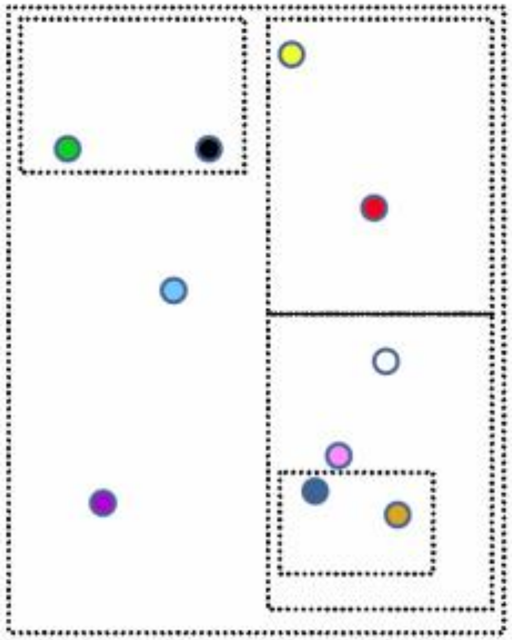
Other Index: BSP Tree (Binary Space Partitioning Tree)





§ 7 Structuring geographic information (cont.)

Other Index: BANG File (Balanced And Nested Grid File)



§ 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
  - geometry and attribute editing
  - database maintenance(e.g., system administration and tuning)
  - creating and updating indexes and topology
  - importing and exporting data
  - georeferencing objects
- These tools form workflow tasks that are exposed within the framework of a WYSIWYG (what you see is what you get) editing environment
- Data entered into the editor must be stored persistently in a file system or database
- Access to the database must be carefully managed to ensure continued security and quality is maintained

§ 9 Conclusion

- Database management systems are now a vital part of large modern operational GIS
- They bring with them standardized approaches for storing and, more importantly, accessing and manipulating geographic data using the SQL query language
- Innovative work in the GIS field has extended standard DBMS to
- store and manage geographic data and has led to the development of long transactions and versioning that have application across several fields