Chapter 4 Spatial data & spatial DB sys

1. Introduction

spatial DB sys is different from conventional DB sys

- stores complex data types – points, lines, polygons
- needs sophisticated spatial operators

2. Definition & classification of spatial data

2.1 Spatial data & pseudo-spatial data

1) spatial data : data of spatial attributes that denote a location / near the surface

two important properties : reference to a geographic space
represent at a variety of geographic scales

two fundamental forms - vector & raster

vector – basic unit is the geographic object, represented by points, lines, polygons
ex. Land use, transportation, forest inventories, land cover

raster – basic spatial unit is a grid cell / pixel, serve as a data store & geographical referencing
size of a single pixel – resolution

2) pseudo-spatial data : describe / related to the characteristics of real world features

ex. street address, demographic characteristics

conversion of pseudo-spatial data into spatial data is a very time consuming & resource intensive

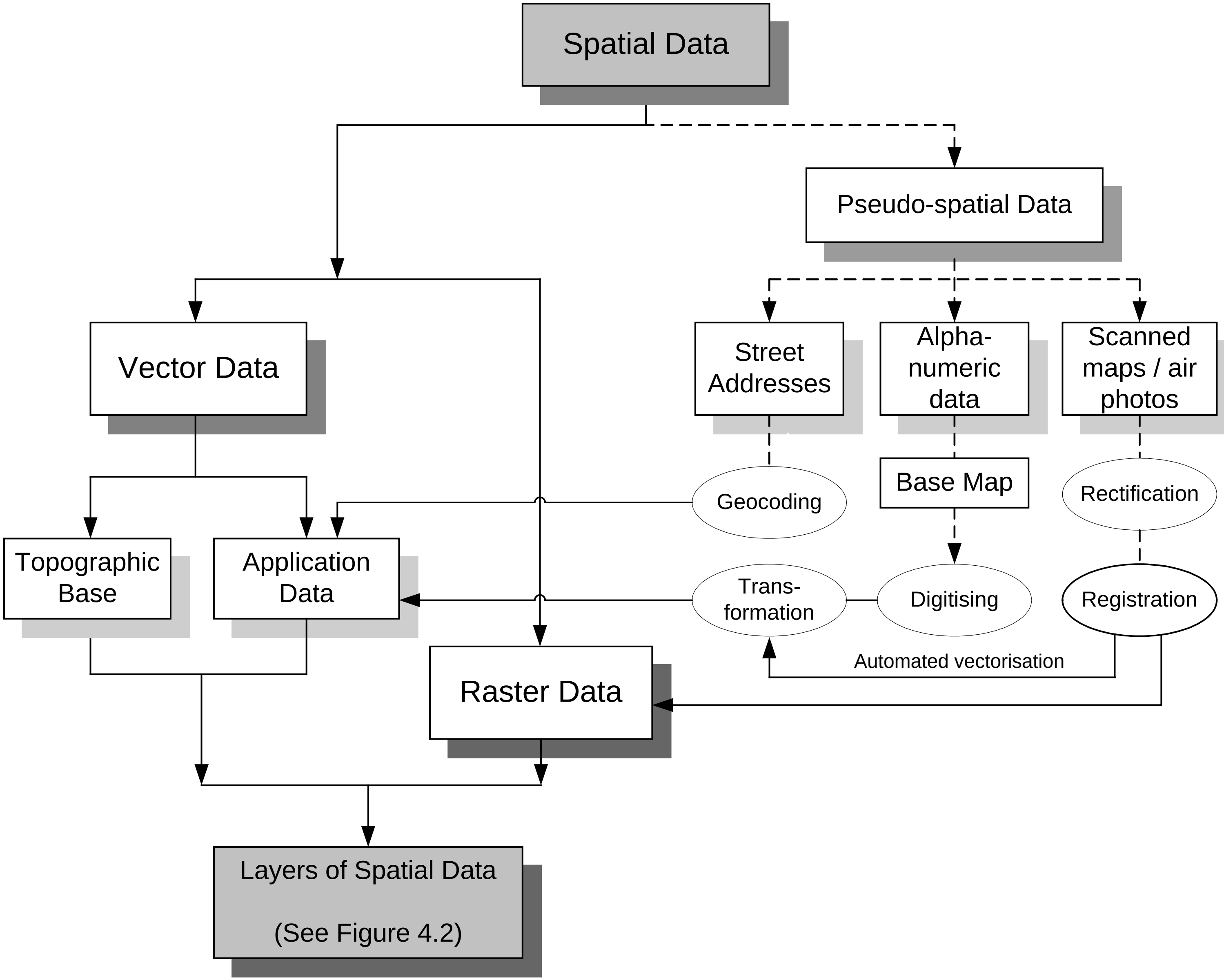


Fig 4-1 Types of spatial data

2. 2 A functional perspective of spatial data

four categories of spatial data in a functional classification

- base map data : geodetic control + various types of topographic base data
- framework data : parcel layer + facilities layer + address layer
- application data : spatial datasets for different DB applications
- business solution : support many operations & decision making functions

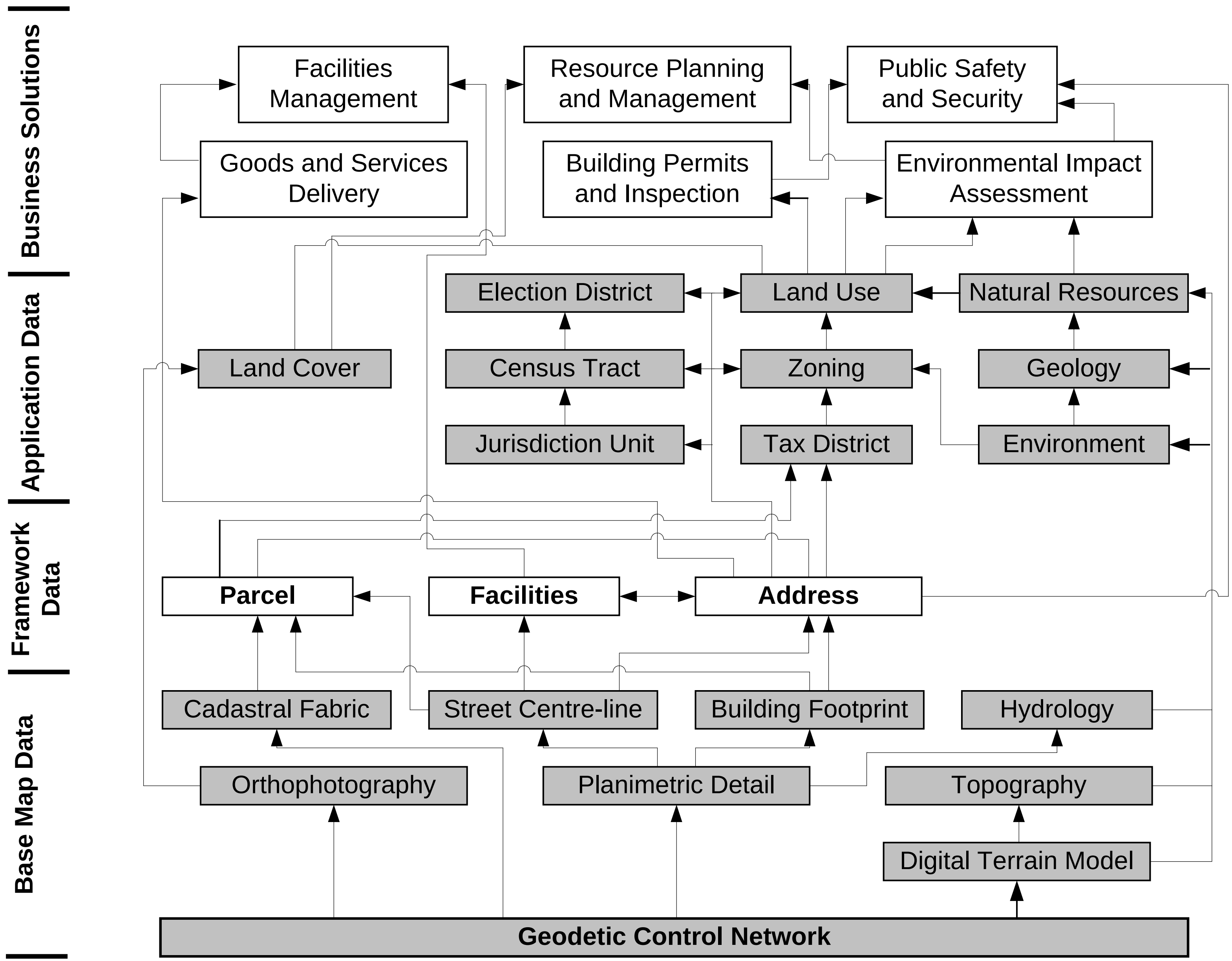


Fig 4-2 A functional classification of spatial data

3. Spatial data structure & DB models

two key aspect of spatial data – geometry + topology

3.1 The concept of a “geometry” of spatial data

geometry is to represent a spatial feature as an object

OGC geometry object model (Open GIS Simple Feature Specification for SQL)

- geometry : non-instantiable construct
- four subclasses (point, curve, surface, geometry collection) : instantiable construct
- there are many geometric types – so called graphical primitives
- geometries sharing the same attributes form a layer (/ feature class)

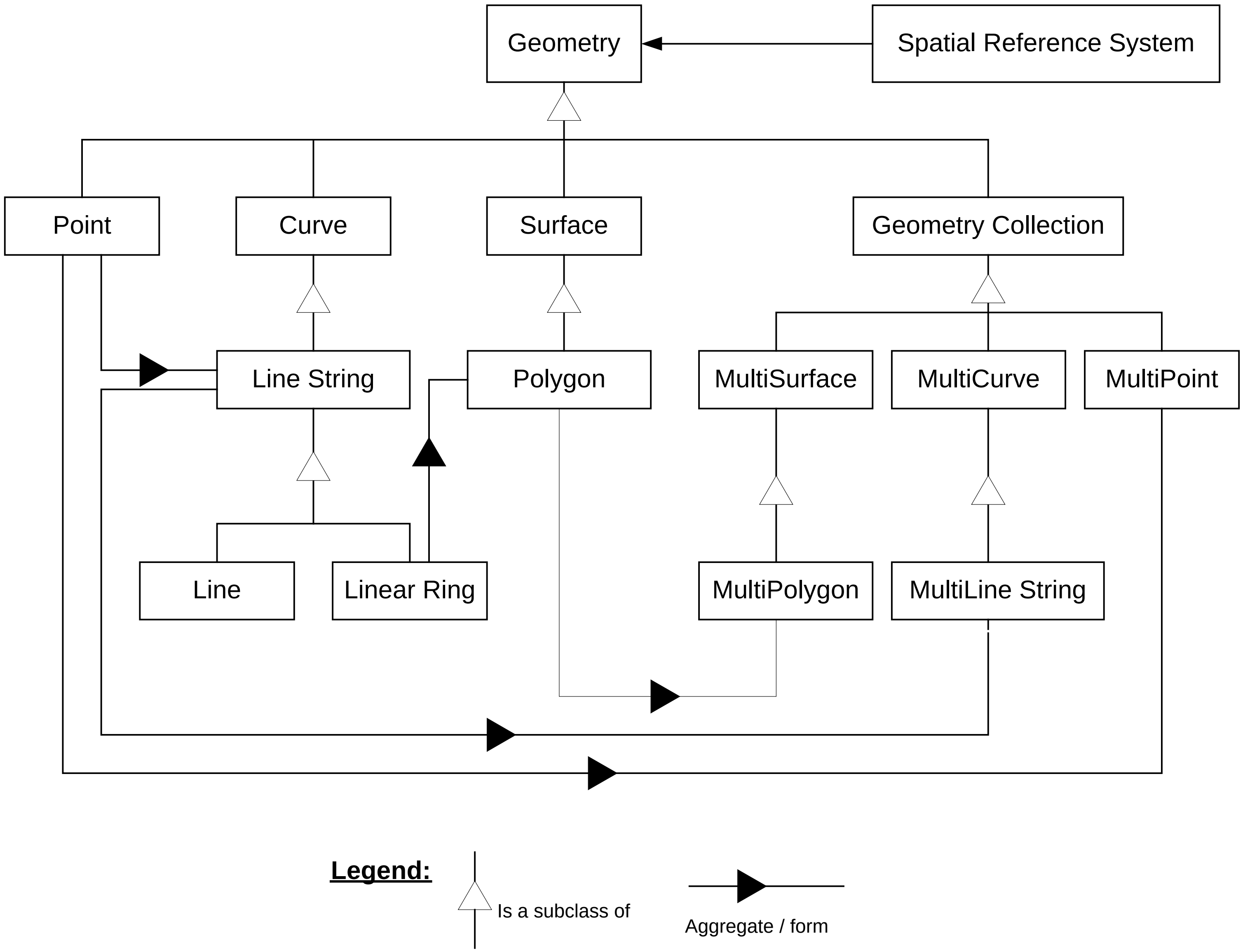
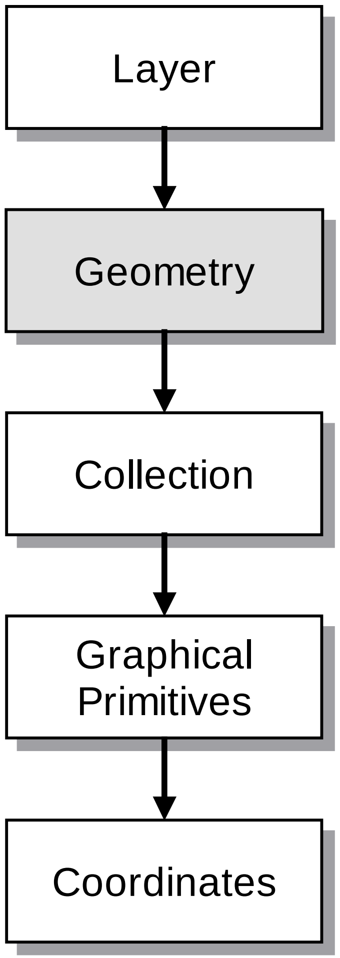


Fig 4-3 The OGC geometry object model

Spatial Hierarchy — Examples



- **Layer**: Roads, rivers, land parcels, land use / cover
- **Geometry**: One or more of the roads, rivers, land parcels, land use / cover polygons from a single layer
- **Collection**: One or more of the road and river segments, parcel lines and land use / cover boundaries in a parent geometry
- **Graphical Primitives**: Individual points, lines and polygons that are used to represent a geometry
- **Coordinates**: (x, y) or (latitude, longitude)

Fig 4-4 The concept of geometry and its relationship w/ other elements as representations of spatial features

3.2 The concept of topology & topological data structures

topology : a field of mathematics studying the properties of geometric figures & their relationships

typically studying adjacency, connectivity, containment

relationship can be defined by 3 primitives – nodes(0cell), edges(1cell), polygons(2cell)

one & two dimensional topology : 1D - network topology by nodes + edges

2D – planar topology by closed polygons

terms definition

node : 0D, one / more edges (ex. arcs, chains, lines) connect to form a topological junction

edge : 1D, formed by a directed, non-branching line segments bounded by a from & to node

polygon : 2D, closed by connected & directed edges

usually, arc-node topology is built to enforce the topological relationship

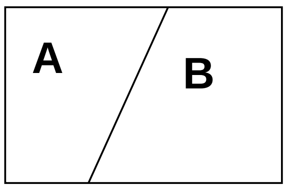
ex. geo-relational data model of an Arc/Info coverage – AAT, PAT tables

advantage of topological data structure

automatic detection & correction of digitizing & editing errors, artifacts

reduction of storage requirements (ex. boundaries shared by adjacent polygons are stored once)

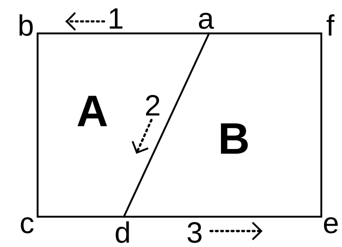
enables sophisticated spatial analysis & applications



Polygon A = (403600, 275700), (403000, 275700), (403000, 275000), (403300, 275000), (403600, 275700)

Polygon B = (403600, 275700), (403300, 275000), (404000, 275700), (404000, 275700)

(a) Non-topological (cartographic) data structure



Polygon File

Poly_ID	Arcs
A	1, 2
B	2, 3

Arc File

Arc_ID	Vertices
1	b,c
2	-
3	e,f

Node File

Node_ID	X	Y
a	403600	275700
d	403300	275000

Coordinate File

Vertice_ID	X	Y
b	403000	275700
c	403000	275000
e	404000	270500
f	404000	275700

Network Topology File

Arc_ID	F_node	T_node
1	a	d
2	d	a
3	d	a

Polygon Topology File

Arc_ID	L_Poly	R_Poly
1	A	World
2	A	B
3	B	World

(b) Topological data structure

Fig 4-5 Topological enforcement & topological data structure

3.3 Non-topological data structure

shapefile – non topological data structure, by ESRI

open data model w/ full-polygon data structure

limitations

- limited cartographic rendering capability
- incompatibility w/ relational DB management principles & techniques
- not efficient spatial analysis
- lack of support for transferring metadata

3.4 The geo-relational model

conventional data model for spatial data

spatial data are abstracted into a series of independently defined layers

each layer represents a selected set of associated spatial features (ex. road, soil type)

spatial features on each layer are the same type of graphical primitives – points, lines, polygons

attributes are stored in separate relational tables – attribute tables

attribute tables are logically linked by the unique feature identifiers (FID)

best example – arc node model (of ESRI coverage)

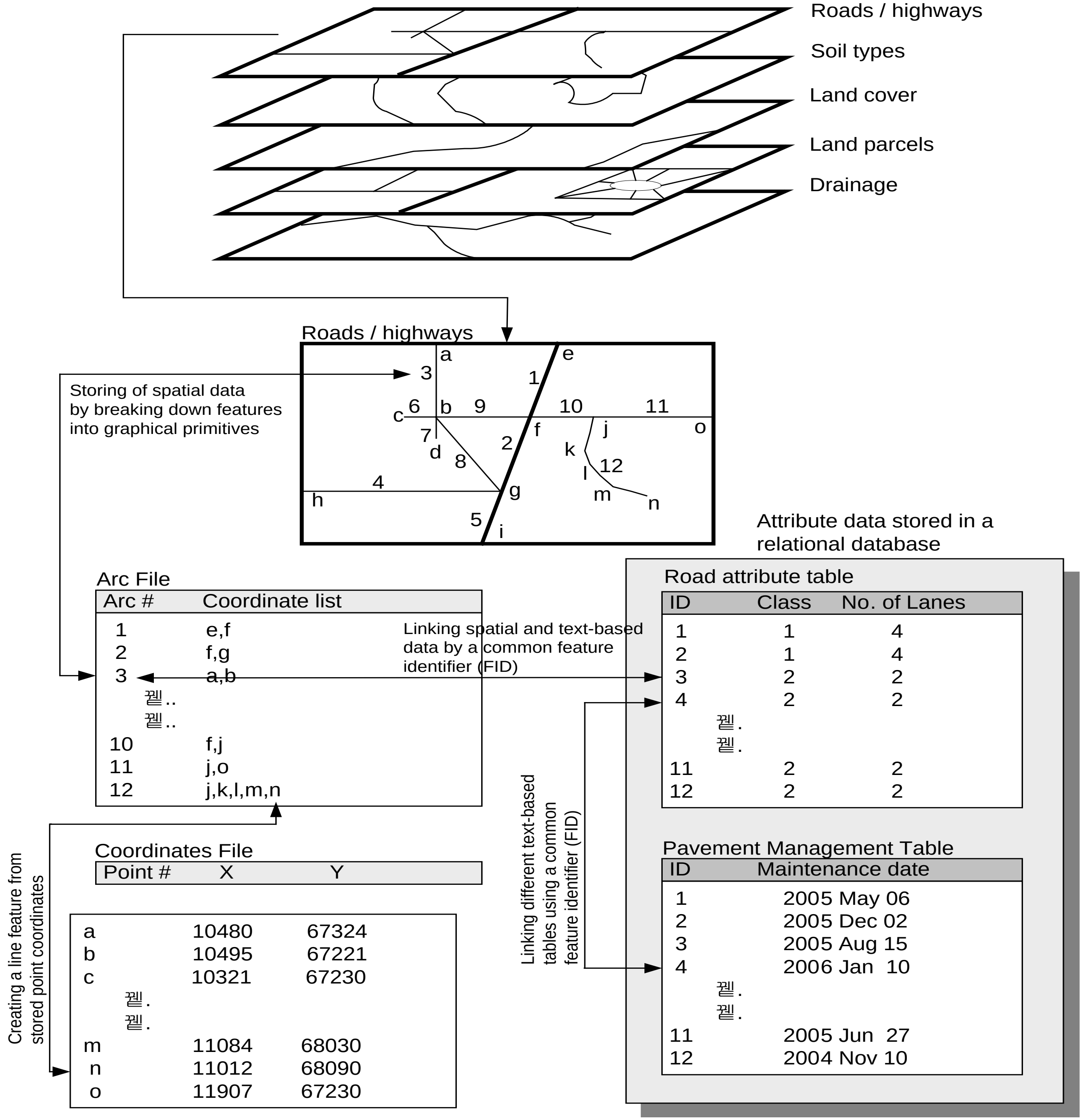


Fig 4-6 The geo-relational model

3.5 The geodatabase model

allows a user to define spatial data as specific abstract data types

→ making it possible to store spatial data w/ their associated attribute data in a single DB

advantages (compare to geo-relational model)

- take full advantage of the available indexing, transaction management
- DB constraint mechanisms to maintain the integrity of the spatial data
- cost effective

best example – geodatabase (of ESRI) – from personal geoDB to enterprise geoDB

- spatial data that share the same attributes are stored in a single table
- table has two sets of fields – predefined fields : FID, geometry, area
  - custom fields : attribute data associated w/ spatial data
    - values can be updated by transaction processing

topology is implemented by using integrity rules stored in a topology tables

- enforce topological relationship to feature class in a single table
- ex. adjacency, connectivity, containment

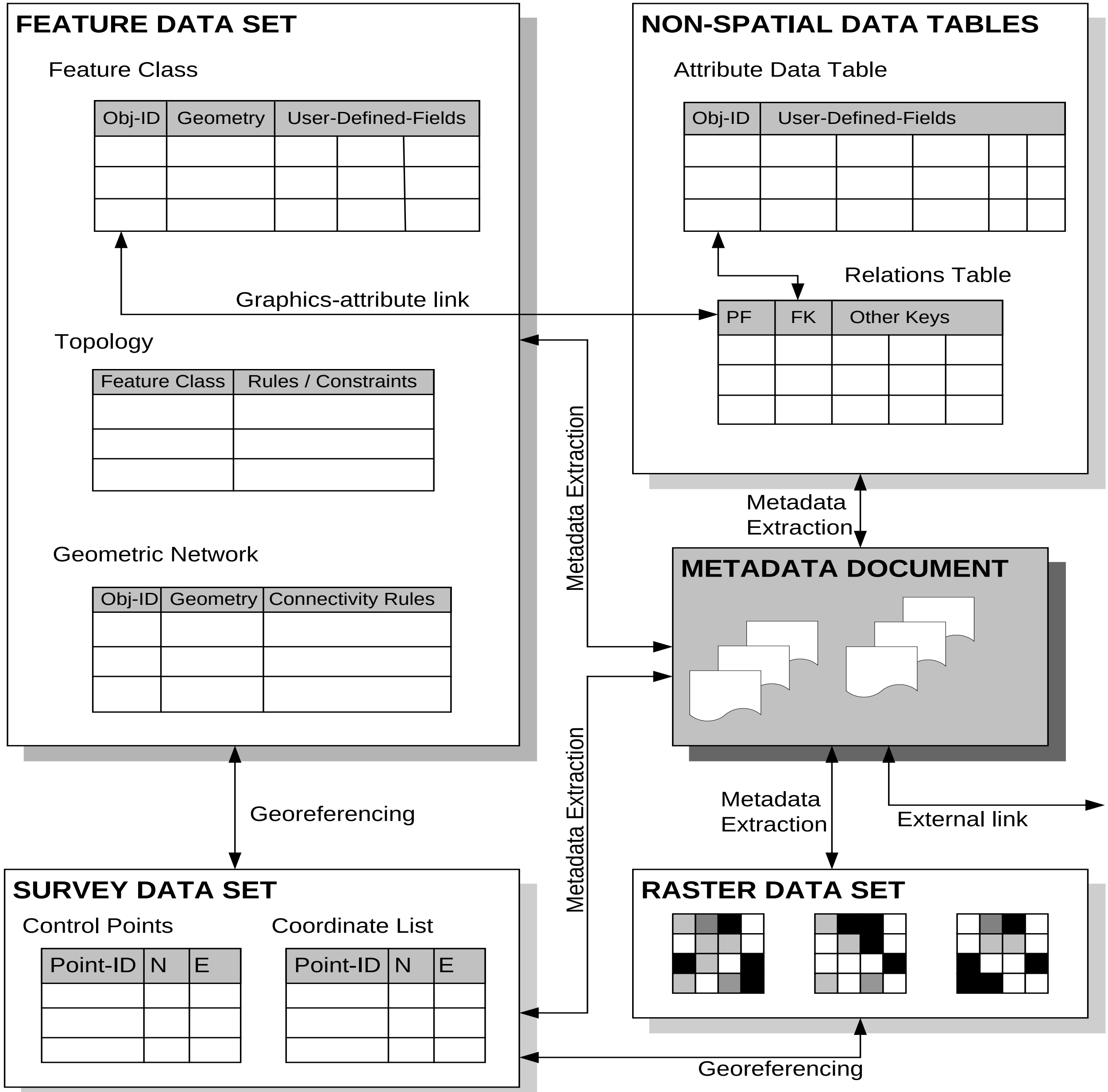


Fig 4-7 Structure of a spatial DB using a DBMS for the storage of spatial data & topological relationships

Predefined fields (FID, Geometry, Shp-area) | **Custom fields** (Owner, SIN, Zoning, Frontage, Date reg.)

FID	Geometry	Shp-area	Owner	SIN	Zoning	Frontage	Date reg.
100		6000.5	VanDamme	004334125	1a	50.0	2003 05 12
102		5600.0	McGrath	200298900	1a	48.5	2003 09 20
124		7200.0	Henderson	434222234	a	60.5	2002 12 12
137		10800.0	Thornley	421004009	1b	200.0	2003 06 01
166		8400.5	Valade	334222090	1a	80.5	2001 05 20

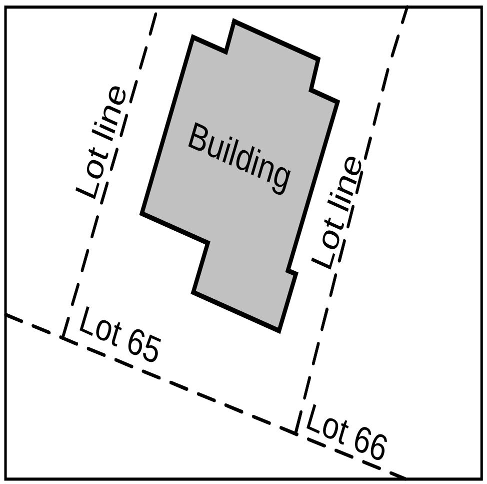
Linking to another relational table using a unique key

SIN	Address	Telephone
004334125	90 Highview Avenue	519-0909
200298900	450 Kingsway East	690-0808
434222234	1234 Hamilton Road	690-2234
421004009	400 Hunt Club Drive	417-8485
334222090	1185 University Street	680-9121

Linking to a lookup table using a classification code

Code	Description
1a	Single-family residential
1b	Multi-family residential
2	Commercial
3	Industrial
4	Institutional

Fig 4-8 Table structure of a geodatabase



Topology File

Feature Class	Rule	Feature Class
Lot_lines	Must not have dangles	
Lots	Must not overlap	
Owner_parcel	Must be closed	
Lot_lines	Must be covered by	Lots
Buildings	Must be covered by	Owner-parcel
Buildings	Must be covered by	Lots
Buildings	Must not overlap	Lot_lines
Lots	Must be formed by	Lot-lines
Lot_lines	Must not overlap	Buildings

Fig 4-9 Storing topological relationships using an integrity rule

4. Spatial DB systems

4.1 Definition & classification of spatial DB systems

three characteristics : a sort of DB sys

offers spatial data type (SDT) & query language

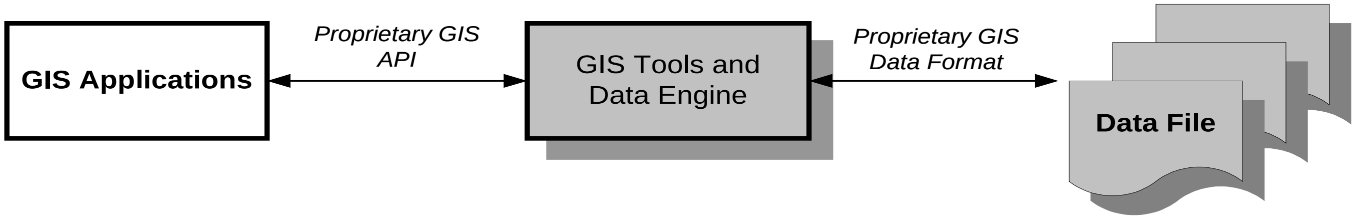
provides at least spatial indexing & efficient algorithms for spatial joins

spatial DB sys works as the underlying technology of GIS

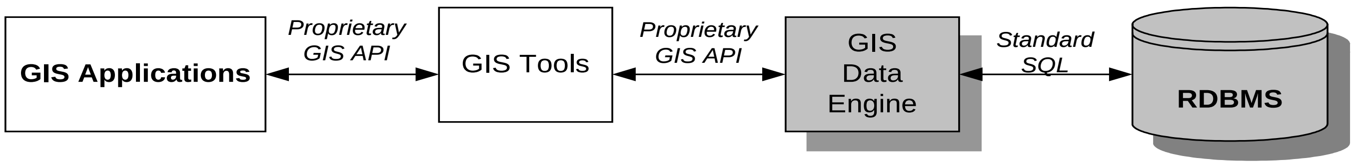
many spatial DBs are now developed as spatial data warehouse

→ supply GIS users w/ timely & relevant spatial data

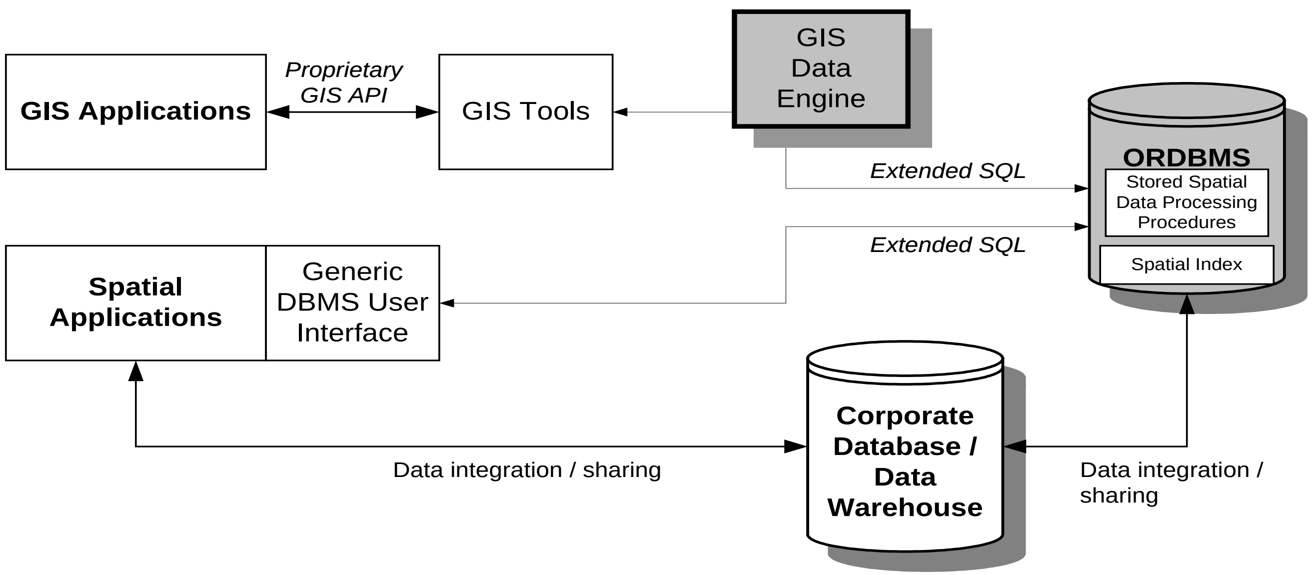
spatial DB sys vs. GIS → Table 4-1



(a) Data file-based spatial data processing using a GIS before the mid-1990s



(b) DBMS-based spatial data processing using a GIS in the late 1990s



(c) Today

뭘 spatial data processi

Fig 4-10 Evolution of spatial data processing

Table 4-1. The division of work between spatial database systems and GIS

Systems	*Primary Tasks*
Geographic Information Systems	o Data Collection and Editing o Data Analysis o Generation of Maps and Cartographic Information Products
Spatial Database Systems	o Data Storage and Management o Spatial Indexing o Data Security and Integrity o Spatial Data Query

4.2 Characteristics of spatial DB systems

4.2.1 Spatial data types (user-defined / abstract data types)

OGC geometry object model provides a conceptual standards-based framework for ADT

different SW vendors implement the concept in different ways

- Oracle Spatial – 9 SDT called geometric primitive types
- IBM DB2 Spatial Extender – geometry type to describe its ADT
- ESRI geodatabase – feature geometry

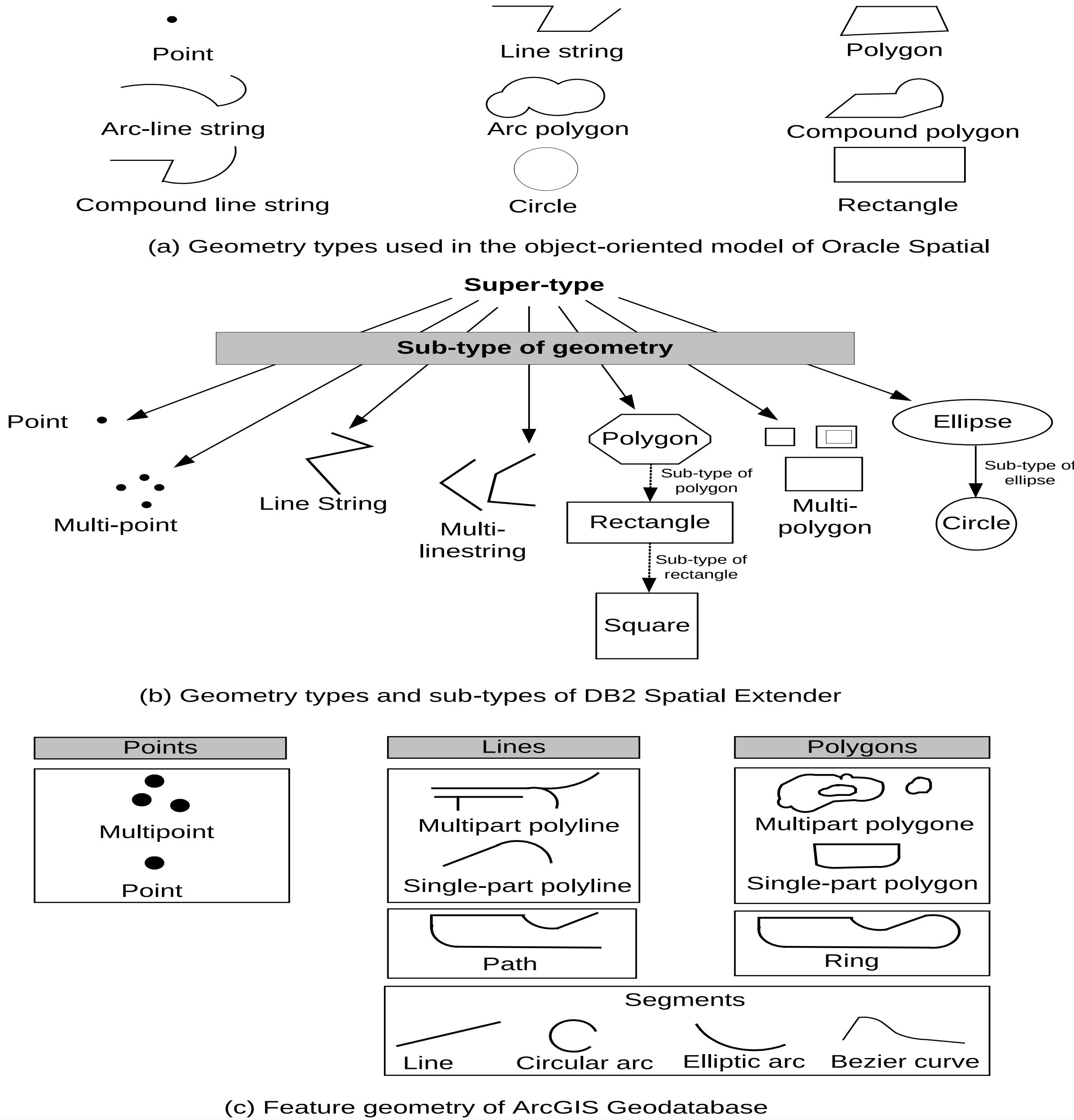


Fig 4-11 Spatial data type used Oracle Spatial, IBM DB2 Spatial Extender & ESRI Geodatabases

4.2.2 Spatial data indexing & access method

spatial indexing – expedite access to & return of data to a user from a DB
more complicated than table indexing – it deals w/ 2D space not linear array in tables

concept is the use of approximation to gradually narrow its search area until objects are found

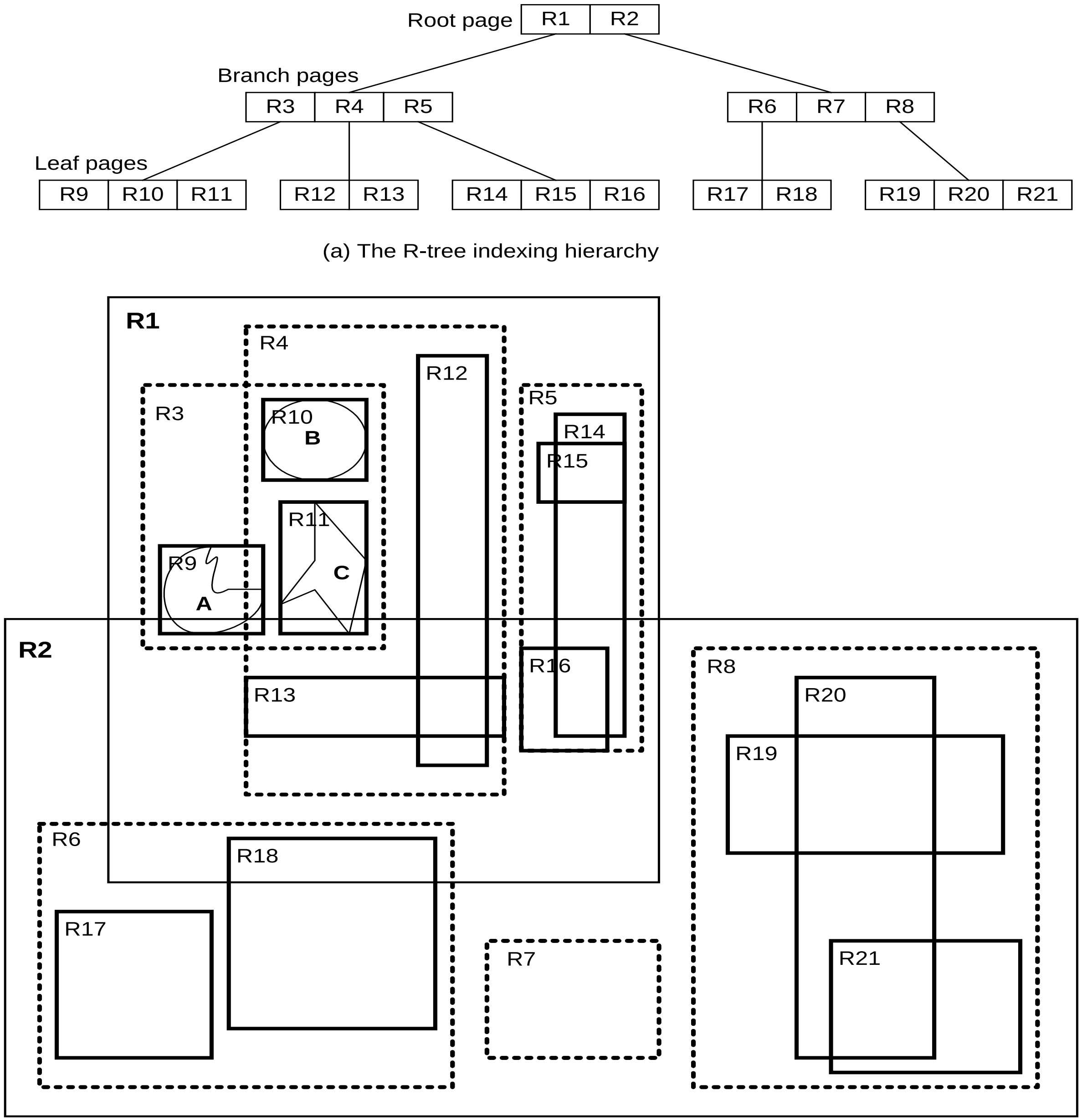
numerous indexing methods – R tree, B tree, quadtree

R-tree : more commonly adopted method

multi-level tree that stores a set of rectangles in each node

* the rectangle - minimum bounding rectangle (MBR)

index stores the reference numbers of the MBRs + coordinates of its 4 corners + object IDs



(b) Spatial relationships among bounding boxes in a R-tree index

Fig 4-12 The R-tree index

4.2.3 Spatial data integrity & constraints

integrity constraints - business rules to protect the data by ensuring accuracy, correctness, validity

six classes of spatial data integrity constraints (Cockcroft's view)

- static topological int conts – ex. all polygons must be closed
- transition topological int conts – ex. if a polygon boundary is modified, all the conjugate polygons must be updated simultaneously
- static semantic int conts – ex. area of land parcel must not be negative
- transition semantic int conts – ex. subdivided land parcels must have the same sum area as the original land parcel
- static user-defined int conts – ex. rivers wider than 2m must be stored as polygons
- transition user-defined int conts – ex. after re-zoning a land parcel, the land use status must be updated within 2 days

cf. data modeling & DB operations case (in Chapter 2)

: domain constraints, key& relation constraints, semantic constraints

4.2.4 Long transaction management

DB transaction - involves movement of data in & out of DB + recording of the process

spatial DB transaction is different from the conventional DB transactions

need to handle long transactions (ex. road fabric update – takes several days, not seconds)

different vendors use different solutions to the long transactions problem

ex. Oracle – workspace manager

: support multiple versions of all records in a table

users can change these versions independently & share w/ others

4.3 Spatial data processing

4.3.1 Classification of spatial operators

a spatial query is formulated using one / more operators

OGC classification of spatial operators (→ Table 4-2)

- basic operators – allow to access the general properties of a geometry
- topological operators – express spatial relationship between geometries
- spatial analysis operators – allow to construct analytical spatial queries using a single /multiple geometries

Table 4-2. OGC spatial operators defined on the class geometry

Classes	***Operators***	***Operator Functions***
Basic Operators	Spatial Reference	Returns the reference system of the geometry
	Envelope	Returns the minimum bounding rectangle of the geometry
	Export	Converts the geometry into a different representation
	IsEmpty	Tests if the geometry is the empty set or not
	IsSimple	Returns TRUE if the geometry is simple
	Boundary	Returns the boundary of the geometry
Topological Operators	Equal	Tests if the geometries are spatially equal
	Disjoint	Tests if the geometries are disjoint
	Intersect	Tests if the geometries intersect
	Touch	Tests if the geometries touch each other
	Cross	Tests if the geometries cross each other
	Within	Tests if a geometry is within another geometry
	Contain	Tests if a given geometry contains another geometry
	Overlap	Tests if a given geometry overlaps another given geometry
	Relate	Returns TRUE if the spatial relationship specified by the 9-Intersection matrix holds
Spatial Analysis Operators	Distance	Returns the shortest distance between any two points of two given geometries
	Buffer	Returns a geometry that represents all points whose distance from the given geometry is less than or equal to a specified distance
	ConvexHull	Returns the convex hull of a given geometry
	Intersection	Returns the intersection of two geometries
	Union	Returns the union of two geometries
	Difference	Returns the difference of two geometries
	SymDifference	Returns the symmetric difference (i.e. the logical XOR) of two geometries

4.3.2 Spatial operations & filtering

large size of a typical spatial DB + complexity of spatial operations → requires filtering

example solution : Oracle Spatial : two-tier approach

- primary filter – reduce the number of candidate geometries by the spatial index of DB
- secondary filter – made up of one /more spatial operators

advantages - expedite the process of accessing the DB

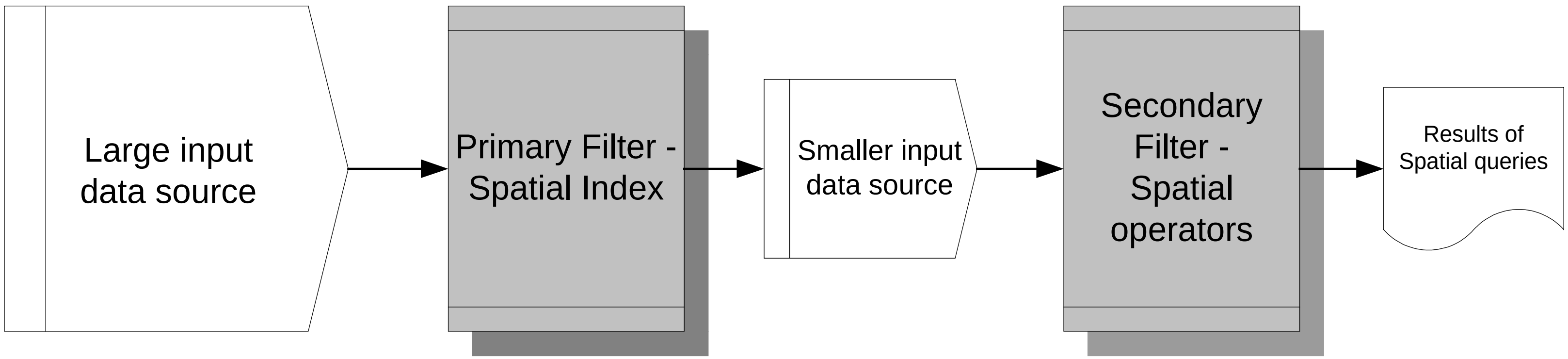


Fig 4-13 Spatial query using the method of two-tier filtering

4.3.3 Topological relations & predicates

among many spatial operators, topological operators play a significant role in spatial queries

topological relations are used w/ **predicates**

* **predicates** = Boolean function, return 1(TRUE) if a comparison meets the function criteria
0(FALSE) otherwise

basic problem w/ the use of topological predicates is to define all possible relationships

DE-9IM (Dimensionally Extended 9 Intersection Model) – define 52 topological relationships
→ too many to be implemented in a spatial DB sys !

thus, most spatial sys include a subset of the possible topological relations (Fig 4-14)

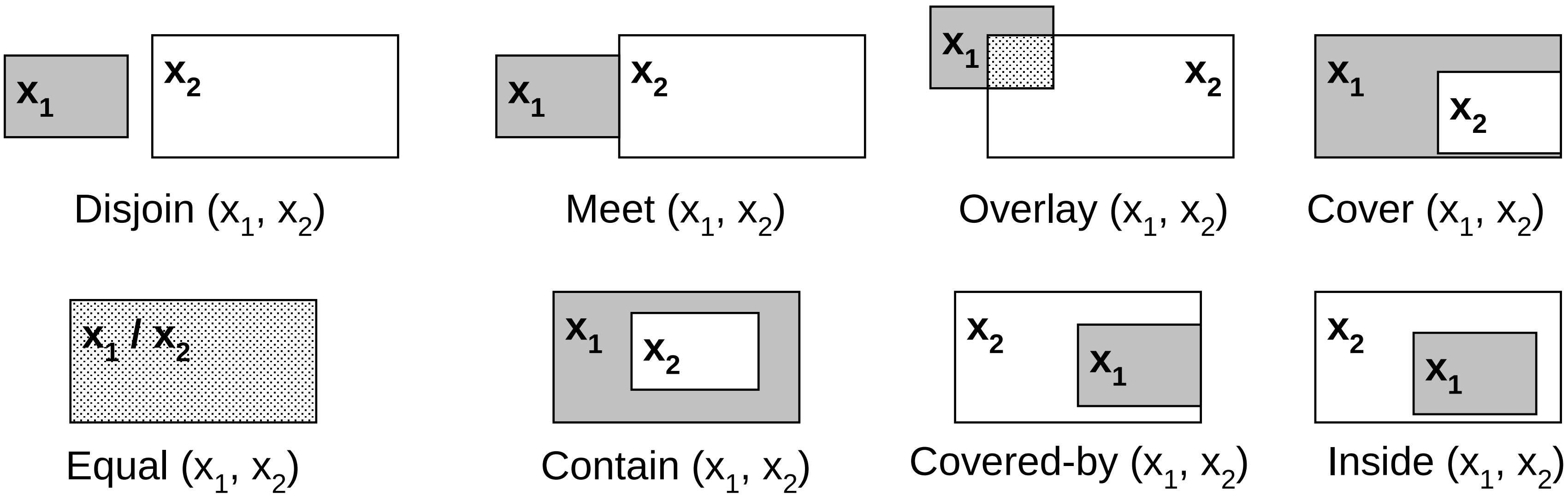


Fig 4-14 The most common topological relations in spatial systems

4.3.4 Spatial joins

spatial join : a spatial query that compares two / more geometries

4.3.5 Spatial SQL

standard SQL is extended by spatial operators (from 1980s)

spatially extended SQL consists of 2 parts : query language + presentation language

- query language – define what data to retrieve
- presentation language – specify how the results of a query are displayed

many spatial extensions to conventional DB sys

- ex. Oracle Spatial, IBM DB2 Spatial Extender

```
SELECT parcel.name
FROM parcel, subdivision
WHERE within (parcel.loc, subdivision.loc)
AND subdivision.name=봠ranebrook?
```

(a) Preservation of the basic SQL SELECT-FROM-WHERE construct

```
CREATE TABLE parcel
        (parcel.ID              char(20)
         geometry               ST_polygon)
```

(b) Defining a spatial object type 밖_polygon? as a spatial data

```
SELECT city
FROM ontario.city
WHERE geometry = PICK;
SELECT city
FROM ontario.city
WHERE city.name = 봫aterloo?
```

(c) Query by location 밝ICK? (by means of a mouse) and by attribute value 봫aterl

```
SET CONTEXT
FOR parcel.geometry
SELECT parcel.geomemtry, building.geometry, road.geometry, easement geometry
FROM parcel, building, road, easement
WHERE parcel.ID = 밚ONDON00221122145678"
```

(d) Setting the background information (building, road, easement) for a spatial object 밖a

```
SET LEGEND
        COLOUR          green
        LINE.TYPE       dashed
FOR SELECT boundary.geometry
FROM parcel
```

(e) Setting the property of a legend

```
SELECT parcel
FROM parcel.layer
WHERE geometry = ZOOM.WINDOW;
SELECT parcel
FROM parcel.layer
WHERE geometry = PICK
```

(f) Restricting a query to a specific area

Fig 4-15 Examples of using spatial SQL