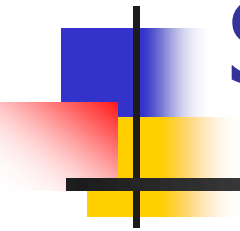
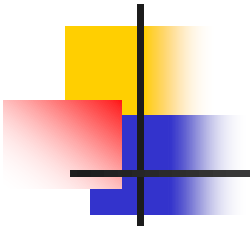
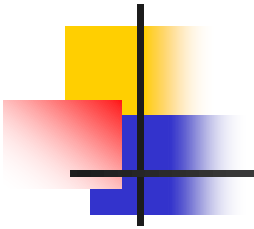


Spatial Database Systems



Spatial Database Applications

- GIS applications (maps):
  - Urban planning, route optimization, fire or pollution monitoring, utility networks, etc
- Other applications:
  - VLSI design, CAD/CAM, model of human brain, etc
- Traditional applications:
  - Multidimensional records

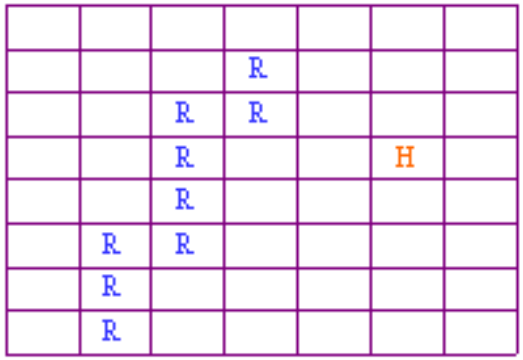


What is a Spatial Database?

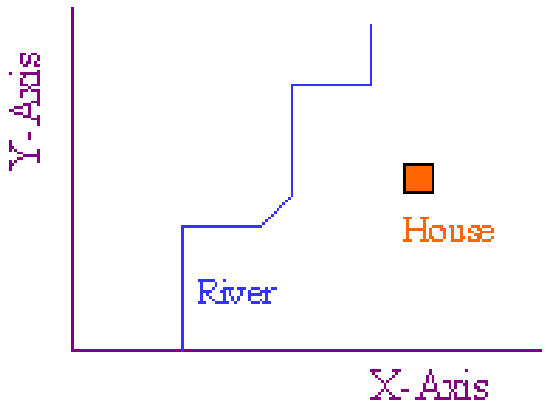
- A SDBMS is a DBMS
- It offers spatial data types/data models/ query language
  - Support spatial properties/operations
- It supports spatial data types in its implementation
  - Support spatial indexing, algorithms for spatial selection and join

Spatial Representation

- Raster model:

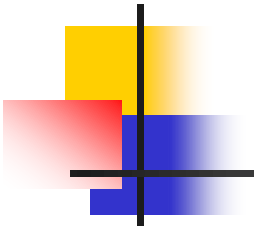


- Vector model:



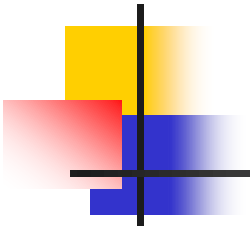
Spatial data types

- Point : 2 real numbers
- Line : sequence of points
- Region : area included inside n-points



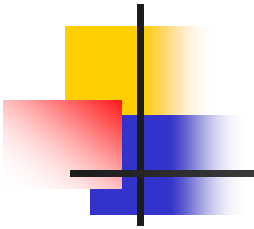
Spatial Relationships

- Topological relationships:
  - adjacent, inside, disjoint, etc
- Direction relationships:
  - Above, below, north_of, etc
- Metric relationships:
  - “distance < 100”
- And operations to express the relationships



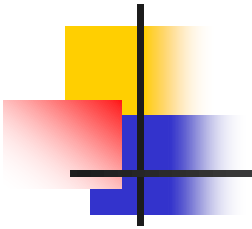
Models, Algebras, Languages

- Extent relational model, or use Object-relational model: define new ADTs
- Spatial algebra: ex. ROSE algebra
- Query languages:
  - Extend SQL : GEOQL, PSQL
  - New graphical languages: GEO-SAL



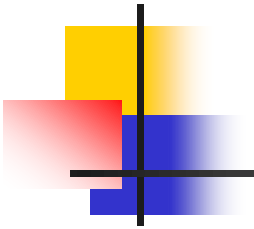
Examples

- A database:
  - Relation states(sname: string, area: region, spop: int)
  - Relation cities(cname: string, center: point; ext: region)
  - Relation rivers(rname: string, route:line)
- SELECT * FROM rivers WHERE route intersects R
- SELECT cname, sname FROM cities, states WHERE center inside area
- SELECT rname, length(intersection(route, California)) FROM rivers WHERE route intersects California



Spatial Queries

- Selection queries: “Find all objects inside query q”, inside-> intersects, north
- Nearest Neighbor-queries: “Find the closets object to a query point q”, k-closest objects
- Spatial join queries: Two spatial relations S1 and S2, find all pairs: {x in S1, y in S2, and x rel y= true}, rel= intersect, inside, etc

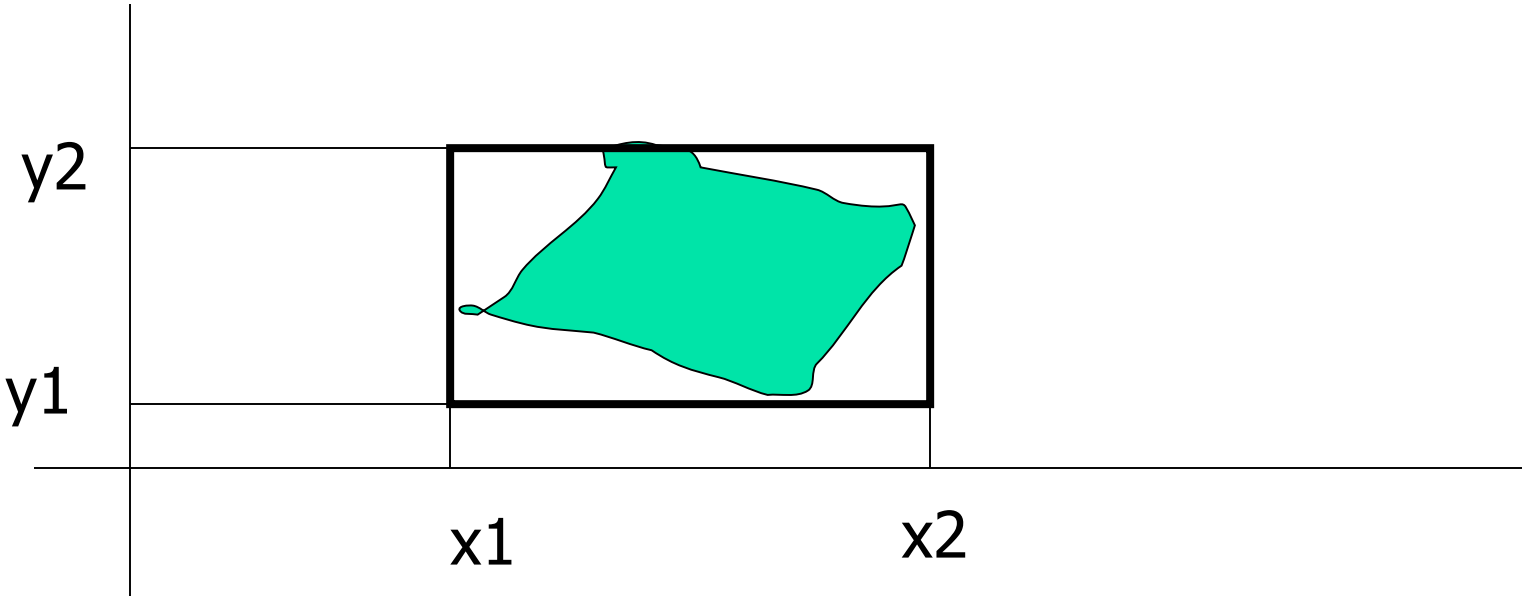


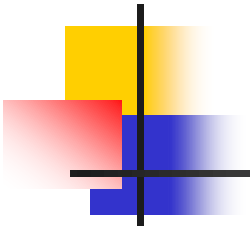
Access Methods

- Point Access Methods (PAMs):
  - Index methods for 2 or 3-dimensional points (k-d trees, Z-ordering, grid-file)
- Spatial Access Methods (SAMs):
  - Index methods for 2 or 3-dimensional regions and points (R-trees)

Indexing using SAMs

- Approximate each region with a simple shape: usually Minimum Bounding Rectangle (MBR) = [(x1, x2), (y1, y2)]

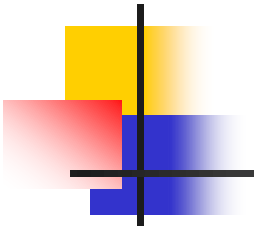




Indexing using SAMs (cont.)

Two steps:

- Filtering step: Find all the MBRs (using the SAM) that satisfy the query
- Refinement step:For each qualified MBR, check the original object against the query



Spatial Indexing

- Point Access Methods (PAMs) vs Spatial Access Methods (SAMs)
- PAM: index only point data
  - Hierarchical (tree-based) structures
  - Multidimensional Hashing
  - Space filling curve
- SAM: index both points and regions
  - Transformations
  - Overlapping regions
  - Clipping methods