GEOPROSTORNE BAZE PODATAKA

GBP–geoprostorne baze podataka - Uvod

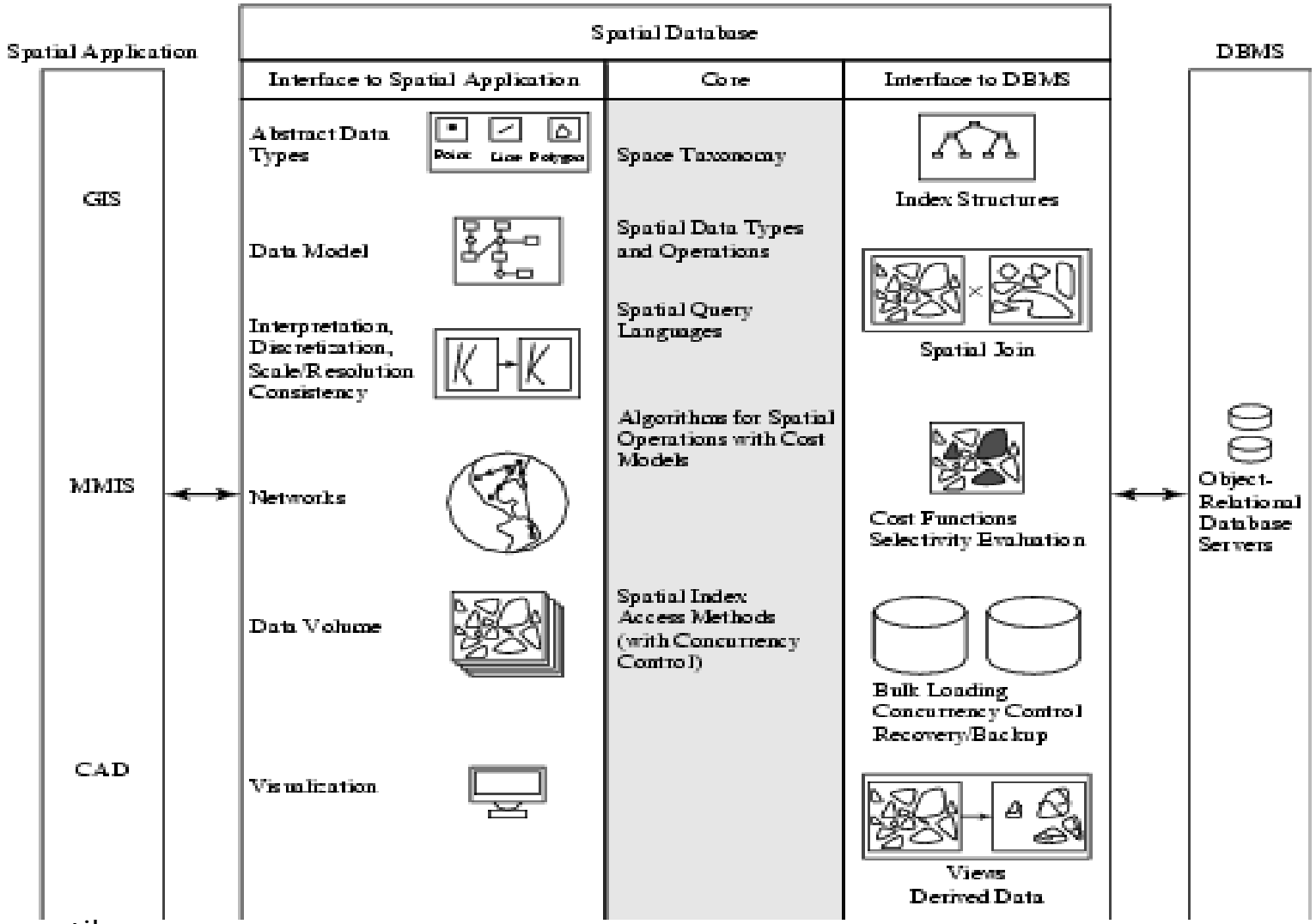
Vrednosti SDBMS

- Tradicionalni (ne-prostorne) sistemi za upravljanje bazama podataka omogućuju:
  - Izdržljivost nasuprot padovima sistema
  - Omogućen konkurentan pristup podacima
  - Skalabilnost prilikom upita nad velikom količinom podataka
  - Uspešnost nad ne-prostornim upitima, za razliku od prostornih upita
- Ne-prostorni upiti:
  - Koja su imena fakulteta koji imaju više od 1000 studenta
- Prostorni upiti:
  - Koja su imena fakulteta koji imaju više od 1000 studenta u krugu od 100km od Beograda

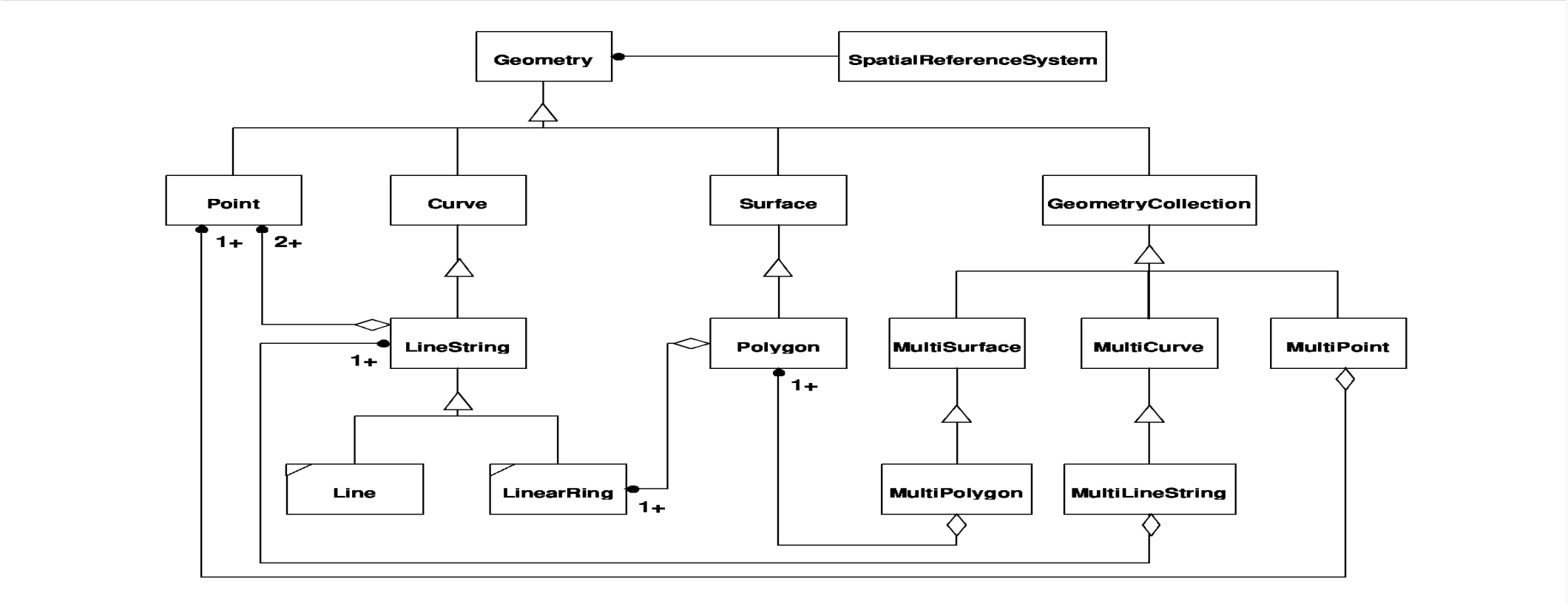
Vrednosti SDBMS – Primeri Prostornih Podataka

- Primeri ne-prostornih podataka
  - Imena, brojevi telefona, email adrese
- Primeri prostornih podataka
  - Popis stanovništva
  - Satelitski snimci - terabyte podataka po danu
  - Vremenski i Klimatski Podaci
  - Reke, Farme, ekološki uticaj
  - Medicinski – rentgen snimci

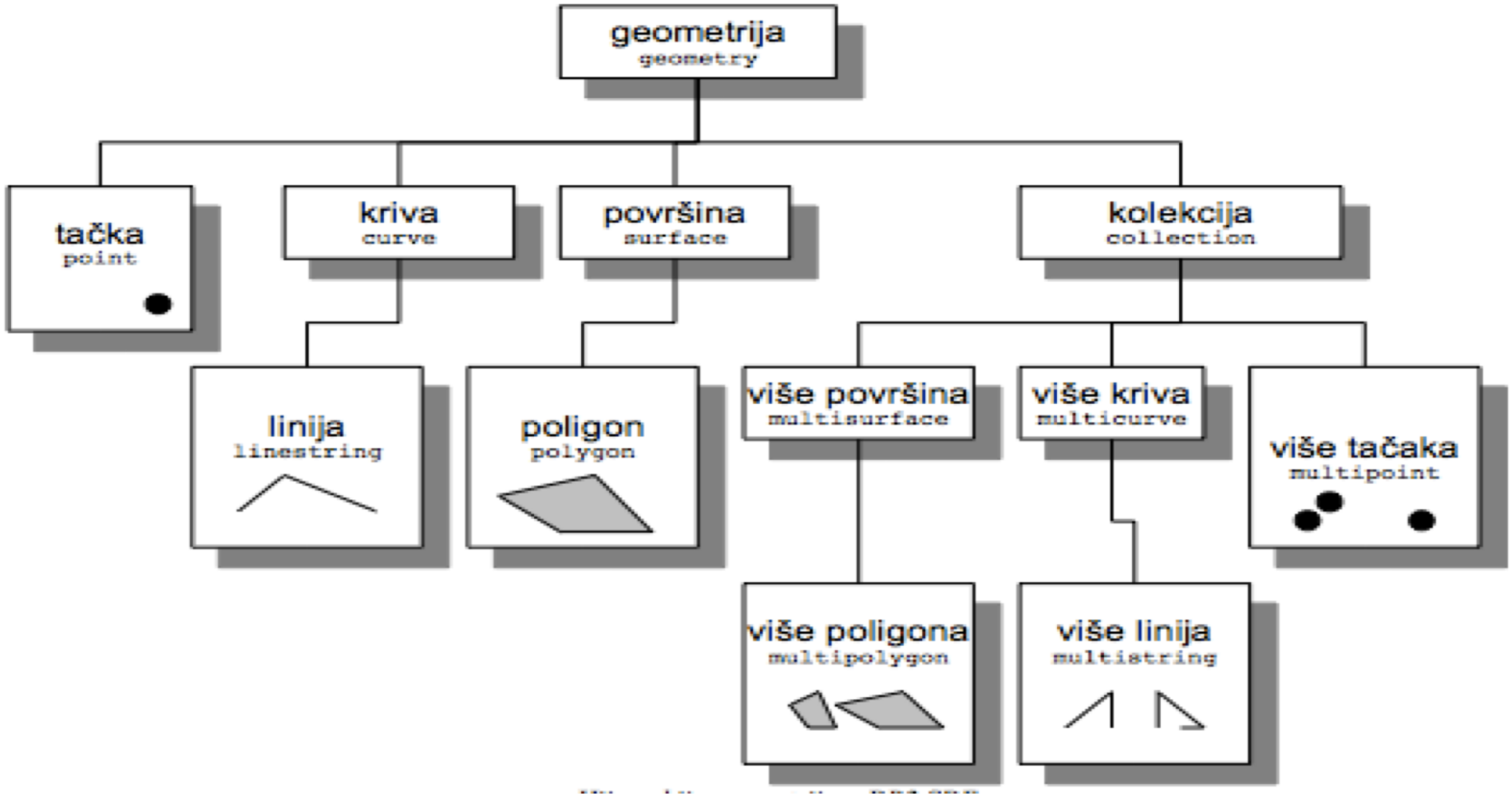
Troslojna Arhitektura



OpenGIS -Tip geometry



OpenGIS -Tip geometry



OpenGIS -Tip geometry

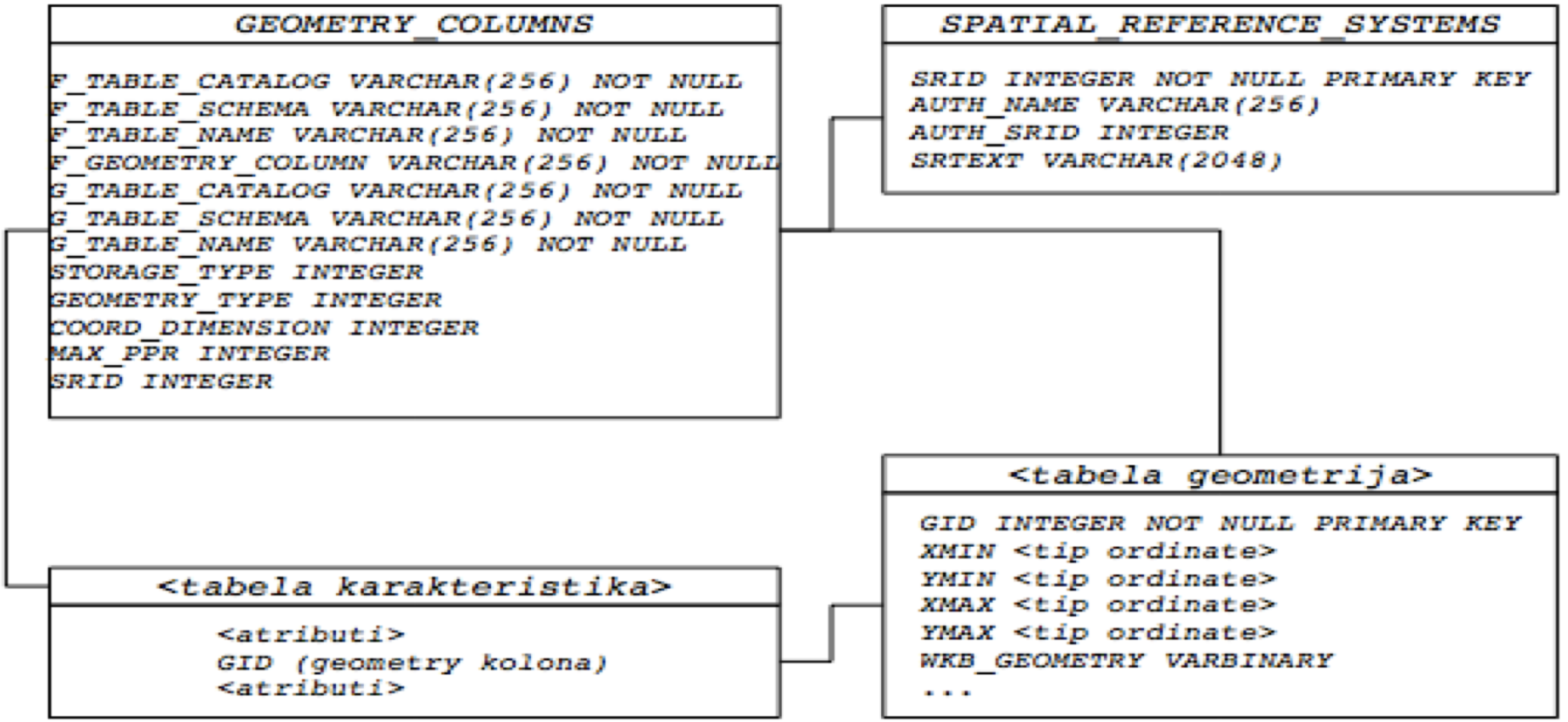
geometry

```
+Dimension():Integer
+GeometryType():String
+SRID():Integer
+Envelope():Geometry
+AsText():String
+AsBinary():Binary
+IsEmpty():Integer
+IsSimple():Integer
+Boundary():Geometry

+Equals(g:Geometry):Integer
+Disjoint(g:Geometry):Integer
+Intersects(g:Geometry):Integer
+Touches(g:Geometry):Integer
+Crosses(g:Geometry):Integer
+Within(g:Geometry):Integer
+Contains(g:Geometry):Integer
+Overlaps(g:Geometry):Integer
+Relate(g:Geometry, pattern:String):Integer

+Distance(g:Geometry):Double
+Buffer(distance:Double):Geometry
+Convexhull():Geometry
+Intersection(g:Geometry):Geometry
+Union(g:Geometry):Geometry
+Difference(g:Geometry):Geometry
+SymDifference(g:Geometry):Geometry
```

OpenGIS -Tip geometry



Osnovne metode klase geometry

+**Dimension**():Integer – inherentna dimenzija objekta koja mora da je manja ili jednaka koordinatnoj dimenziji (pod dimenzijom podrazumevamo broj nezavisnih veličina, a ne fizičku veličinu objekta !)

+**GeometryType**():String – ime (podklase) geometrije u tekstualnom formatu, tj. string reprezentaciji

+**SRID**():Integer – ID objekta unutar prostornog referentnog sistema

+**Envelope**():Geometry – najmanji pravouganonik u kojem je objekat sadržan (minimum bounding box), definisan sa ((MINX),(MINY),(MAXX),(MAXY)).

+**AsText**():String – geometrija objekta predstavljena u WKT formatu

+**AsBinary**():Binary – geometrija objekta predstavljena u WKB

+**IsEmpty**():Integer – vraća 1 (true) ako je ovo prazna geometrija (sadrži objekat sa 0 tačaka).

+**IsSimple**():Integer – vraća 1 (true) ako geometrija objekta ne sadrži linije koje se prepliću ili dodiruju druge linije ili tačke ovog objekta.

+**Boundary**():Geometry

Metode za proveru prostornih odnosa između geometrija

+**Equals**(g:Geometry):Integer – vraća 1 ako je geometrija objekta prostorno jednaka geometriji g

+**Disjoint**(g:Geometry):Integer – vraća 1 ako je geometrija objekta prostorno nije jednaka geometriji g

+**Intersects**(g:Geometry):Integer – vraća 1 ako geometrija objekta prostorno preseca geometriju g

+**Touches**(g:Geometry):Integer – vraća 1 ako geometrija objekta prostorno dodiruje geometriju g

+**Crosses**(g:Geometry):Integer – vraća 1 ako geometrija objekta prostorno preseca geometriju g

+**Within**(g:Geometry):Integer – vraća 1 ako je geometrija objekta prostorno unutar geometrije g

+**Contains**(g:Geometry):Integer – vraća 1 ako je geometrija objekta prostorno sadrži geometriju g

+**Overlaps**(g:Geometry):Integer – vraća 1 ako je geometrija objekta prostorno preklapa geometriju g

+**Relate**(g:Geometry, pattern:String):Integer – vraća 1 ako je geometrija objekta u prostornoj relaciji sa geometrijom g na osnovu kriterijuma zadatih u stringu pattern proveravajući za presecanja unutrašnjosti, ivica i spoljašnosti geometrija

Metode za prostornu analizu

+**Distance**(g:Geometry):Double – vraća najmanju moguću razdaljinu između dve proizvoljne tačke iz geometrije g i geometrije objekta, računajući u prostornom referentnom sistemu objekta

+**Buffer**(distance:Double):Geometry – vraća geometriju koja sadrži sve tačke čija je razdaljina od proizvoljne tačke geometrije objekta manja od distance

+**Convexhull**():Geometry – vraća geometriju objekta koja je pretvorena u konveksnu površinu

+**Intersection**(g:Geometry):Geometry – vraća geometriju koja sadrži tačke koje su presek geometrije objekta i geometrije g

+**Union**(g:Geometry):Geometry – vraća geometriju koja je unija geometrije objekta i geometrije g

+**Difference**(g:Geometry):Geometry – vraća geometriju koja je razlika geometrije objekta i geometrije g

+**SymDifference**(g:Geometry):Geometry – vraća geometriju koja je simetrična razlika geometrije objekta i geometrije g

GeometryCollection

- GeometryCollection je geometrija koja se sastoji od jedne ili više drugih geometrija.
- Sve geometrije sadržane u objektu ove klase moraju biti u istom prostornom referentnom sistemu u kojem je i GeometryCollection.
- Metode:
  - **NumGeometries**():Integer - vraća broj geometrija sadržanih u ovom objektu
  - **GeometryN**(N:integer):Geometry – vraća n-tu geometriju iz ovog objekta

Point

- Point je bezdimenziona geometrija i predstavlja tačku u kordinatnom prostoru.
- Point ima x kordinatnu vrednost i y kordinatnu vrednost. Ivica objekta klase Point je prazan skup.
- Metode:
  - **X**():Double - vrednost x kordinate
  - **Y**():Double - vrednost y kordinate

MultiPoint

- MultiPoint je bezdimenzioni geometrijski skup i sadrži isključivo objekte klase Point, koji nisu povezani i čiji redosled je proizvoljan

Curve

- Curve je kriva, jednodimenziona geometrija predstavljena nizom tačaka. Podtipovima je moguće definisati kakvu interpolaciju krive želimo.

- Metode:
  - **Length**():Double - Dužina krine u svom prostornom referentnom sistemu
  - **StartPoint**():Point - Početna tačka krive
  - **EndPoint**():Point - Poslednja tačka krive
  - **IsClosed**():Integer - Vraća 1 (TRUE) ako je početna tačka jednaka završnoj
  - **IsRing**():Integer - Vraća 1 (TRUE) ako je početna tačka jednaka završnoj i kriva ne preseca ili dodiruje samu sebe

LineString, Line, LinearRing

- LineString je podtip Curve sa linearnon interpolacijom između tačaka, svaki uzastopni par tačaka definiše dodatni segment linije.
- Line je LineString koji se sastoji od dve tačke.
- LinearRing je linija za koju važi da je početna tačka jednaka završnoj i ne preseca ili dodiruje samog sebe.
- Metode:
  - **NumPoints**():Integer - Broj tačaka u objektu
  - **PointN**():Point - Vraća n-tu tačku objekta.

MultiCurve

- MultiCurve je skup geometrija krivih (Curve). Predstavlja apstraktnu klasu.
- Metode:
  - **Length**():Double - Ukupna dužina krivih u svom prostornom referentnom sistemu
  - **IsClosed**():Integer - Vraća 1 (TRUE) ako je svaka početna tačka jednaka završnoj u svakoj od sadržanih geometrija

MultiLineString

- Podtip MultiCurve koji sadrži geometrije tipa LineString

Surface

- Surface je apstraktna dvodimenziona geometrijska klasa za površinu.

- Metode:
  - **Area**():Double - Površina geometrije u svom prostornom referentnom sistemu
  - **Centroid**():Point - Vraća tačku koja obeležava centroid ove geometrije
  - **PointOnSurface**():Point - Vraća proizvoljnu tačku koja je sadržana u ovoj geometriji

Polygon

- Podtip Surface, definisan jednim spoljašnjim ograničenjem (granicom) i sa 0 ili više unutrašnjih ograničenja. Svaka unutrašnja granica predstavlja rupu u geometriji.

- Metode:
  - **ExteriorRing**():LineString - Vraća spoljašnje ograničenje
  - **NumInteriorRing**():Integer - Vraća broj unutrašnjih ograničenja
  - **InteriorRing**(N:Integer):LineString - Vraća n-to unutrašnje ograničenje

MultiSurface

- MultiSurface je apstraktni dvodimenzionalni skup Surface geometrija.

- Metode:
  - **Area**():Double - Ukupna površina sadržanih geometrija u svom prostornom referentnom sistemu
  - **Centroid**():Point - Vraća tačku koja obeležava centroid geometrija
  - **PointOnSurface**():Point - Vraća proizvoljnu tačku koja je sadržana u ovoj geometriji

MultiPolygon

- Podtip MultiSurface koji sadrži geometrije tipa Polygon

WKT – Well Known Text

- Well known text (WKT) je reprezentacija geometrija koja se može koristiti za konstrukciju novih instanci geometrija ili jednostavno prikazati neku geometriju u alfanumerickom formatu

WKT – Well Known Text

<Kvalifikovani opis geometrije> :=
<Kvalifikovani opis tipa Point> |
<Kvalifikovani opis tipa LineString> |
<Kvalifikovani opis tipa Polygon> |
<Kvalifikovani opis tipa MultiPoint> |
<Kvalifikovani opis tipa MultiLineString> |
<Kvalifikovani opis tipa MultiPolygon> |
<Kvalifikovani opis tipa GeometryCollection>

<Kvalifikovani opis tipa Point> := POINT <Point Text>

<Kvalifikovani opis tipa LineString> := LINESTRING <LineString Text>

<Kvalifikovani opis tipa Polygon> := POLYGON <Polygon Text>

<Kvalifikovani opis tipa MultiPoint> := MULTIPOINT <Multipoint Text>
<Kvalifikovani opis tipa MultiLineString> := MULTILINESTRING <MultiLineString Text>
<Kvalifikovani opis tipa MultiPolygon> := MULTIPOLYGON <MultiPolygon Text>
<Kvalifikovani opis tipa GeometryCollection> := GEOMETRYCOLLECTION <GeometryCollection Text>
<Point Text> := EMPTY | (<Point>)
<Point> := <x> <y>
<x> := literal dvostruke tačnosti
<y> := literal dvostruke tačnosti
<LineString Text> := EMPTY | (<Point > {, <Point > }*)
<Polygon Text> := EMPTY | (<LineString Text > {, < LineString Text > }*)
<Multipoint Text> := EMPTY | (<Point Text > {, <Point Text > }*)
<MultiLineString Text> := EMPTY | (<LineString Text > {, < LineString Text > }*)
<MultiPolygon Text> := EMPTY | (< Polygon Text > {, < Polygon Text > }*)
<GeometryCollection Text> := EMPTY | (<Geometry Tagged Text> {, <Geometry Tagged Text> }*)

Funkcije za rad sa WKT geometrijama

- **GeomFromText**(geometryTaggedText String, SRID Integer) : Geometry Pretvara WKT reprezentaciju u geometry supertip.
- **PointFromText**(pointTaggedText String, SRID Integer): Point
- **LineFromText**(lineStringTaggedText String, SRID Integer) : LineString
- **PolyFromText**(polygonTaggedText String, SRID Integer): Polygon
- **MPointFromText** (multiPointTaggedText String, SRID Integer): MultiPoint
- **MLineFromText**(multiLineStringTaggedText String, SRID Integer): MultiLineString
- **MpolyFromText**(multiPolygonTaggedText String, SRID Integer): MultiPolygon
- **GeomCollFromTxt**(geometryCollectionTaggedText String, SRID Integer): GeomCollection

Primer

- WKT predstava geometrija

  POINT (10, 10)
  LINESTRING (10 10, 20 20, 30 40)
  POLYGON ((10 10, 10 20, 20 20, 20 15)
  MULTIPOINT (10 10, 20 20)
  MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))
  MULTIPOLYGON (((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))
  GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))

- WKT u SQL upitu

  INSERT INTO Countries (Name, Location) VALUES ('Novi Sad' ,
  PolygonFromText('POLYGON((x1 y1, x2 y2, x3 y3, ... , xn yn))', 14);

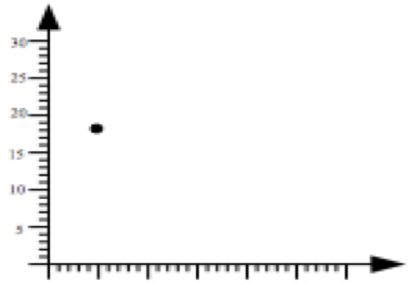
WKB - Well Known Binary

- Well known binary (WKB) je reprezentacija geometrija u binarnoj formi koja je pogodna za skladištenje i pretragu.
- Unutar WKB-a se koriste binarna predstava dva tipa: Unsigned Integer i Double

- **GeomFromWKB**(WKBGeometry Binary, SRID Integer) : Geometry - Pretvara WKT reprezentaciju u geometry supertip.
- **PointFromWKB**(WKBPoint Binary, SRID Integer): Point
- **LineFromWKB**(WKBLineString Binary, SRID Integer) : LineString
- **PolyFromWKB**(WKBPolygon Binary, SRID Integer): Polygon
- **MPointFromWKB** (WKBMultiPoint Binary, SRID Integer): MultiPoint
- **MLineFromWKB**(WKBMultiLineString Binary, SRID Integer): MultiLineString
- **MpolyFromWKB**(WKBMultiPolygon Binary, SRID Integer): MultiPolygon
- **GeomCollFromWKB**(WKBGeometryCollection Binary, SRID Integer): GeomCollection

WKT i WKB - Primer

Tačka na kordinatama (5, 18)



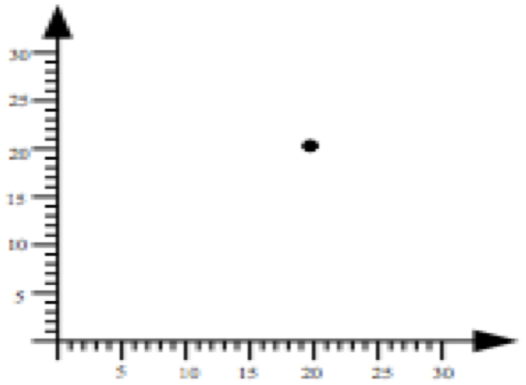
POINT(5 18)

00	00	00	00	01	40	14	00	00	00	00	00	00	40	32	00	00	00	00	00	00
byte order	int32 WKB type = 1				double X = 5								double Y = 18							

ista ova geometrija prestavljena u NDR (little endian) formatu izgleda ovako:

01	01	00	00	00	00	00	00	00	00	00	14	40	00	00	00	00	00	00	32	40
byte order	int32 WKB type = 1				double X = 5								double Y = 18							

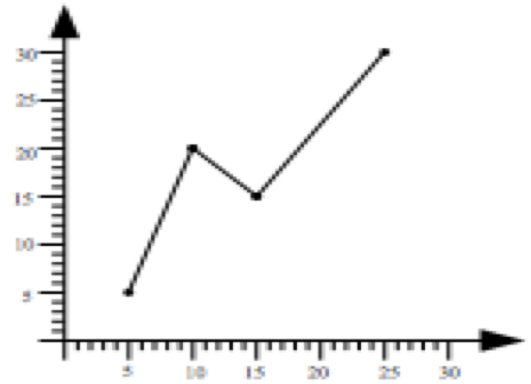
WKT i WKB - Primer



POINT(20 20)

00	00	00	00	01	40	3E	00	00	00	00	00	00	40	3E	00	00	00	00	00	00

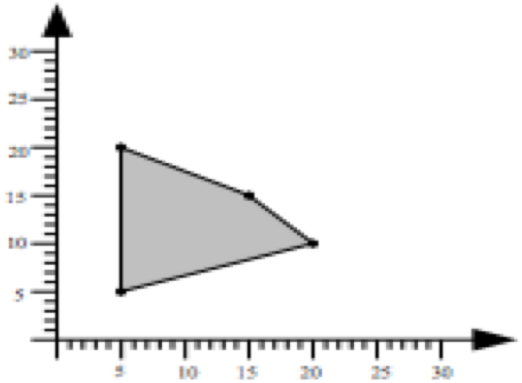
WKT i WKB - Primer



LINESTRING(5 5, 10 20, 15 15, 25 30)

00	00	00	00	02	00	00	00	04	40	14	00	00	00	00	00	00	40	14	00	00	00	00	00
00	40	24	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	2E	00	00	00	00	00
00	40	2E	00	00	00	00	00	00	40	39	00	00	00	00	00	00	40	3E	00	00	00	00	00
00																							

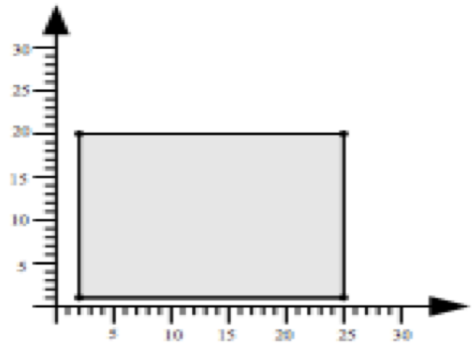
WKT i WKB - Primer



POLYGON((5 5, 5 20, 15 15, 20 10, 5 5))

00	00	00	00	03	00	00	00	01	00	00	00	05	40	14	00	00	00	00	00	00	40	14	00
00	00	00	00	00	40	14	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	2E	00
00	00	00	00	00	40	2E	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	24	00
00	00	00	00	00	40	14	00	00	00	00	00	00	40	14	00	00	00	00	00	00			

WKT i WKB - Primer



POLYGON((2 1, 25 1, 25 20, 2 20, 2 1))

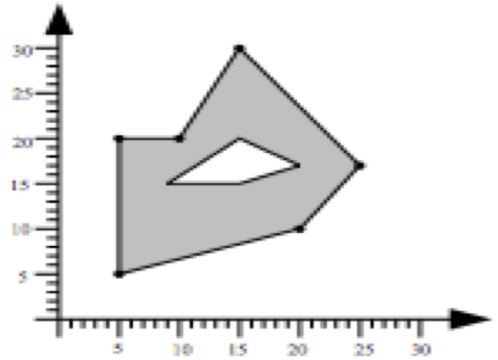
00	00	00	00	03	00	00	00	01	00	00	00	05	40	00	00	00	00	00	00	00
byte order	int32 WKBType = 3				int32 numrings = 1				int32 numPoints = 5				double X = 2							

3F	F0	00	00	00	00	00	00	40	39	00	00	00	00	00	00	3F	F0	00	00	00	00	00	00
double Y = 1								double X = 25								double Y = 1							

40	39	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	00	00	00	00	00	00	00
double X = 25								double Y = 20								double X = 2							

40	34	00	00	00	00	00	00	40	00	00	00	00	00	00	00	3F	F0	00	00	00	00	00	00
double Y = 20								double X = 2								double Y = 1							

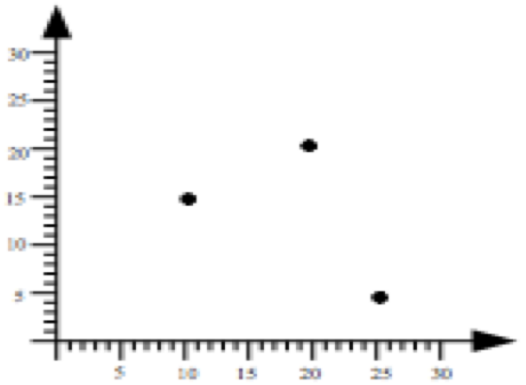
WKT i WKB - Primer



POLYGON((5 5, 5 20, 10 20, 15 30, 25 17, 20 10, 5 5),(10 15, 15 20, 20 17, 15 15, 10 15))

00	00	00	00	03	00	00	00	02	00	00	00	07	40	14	00	00	00	00	00	00	40	14	00
00	00	00	00	00	40	14	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	24	00
00	00	00	00	00	40	34	00	00	00	00	00	00	40	2E	00	00	00	00	00	00	40	3E	00
00	00	00	00	00	40	39	00	00	00	00	00	00	40	31	00	00	00	00	00	00	40	24	00
00	00	00	00	00	40	34	00	00	00	00	00	00	40	14	00	00	00	00	00	00	40	14	00
00	00	00	00	00	00	00	00	05	40	24	00	00	00	00	00	00	40	2E	00	00	00	00	00
00	40	2E	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	34	00	00	00	00	00
00	40	39	00	00	00	00	00	00	40	2E	00	00	00	00	00	00	40	2E	00	00	00	00	00
00	40	24	00	00	00	00	00	00	40	2E	00	00	00	00	00	00	00						

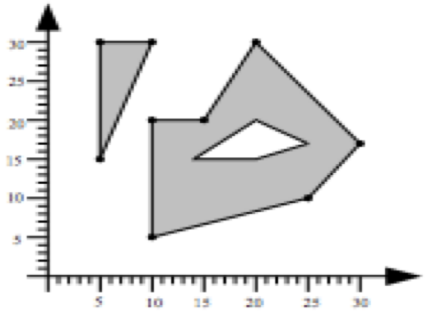
WKT i WKB - Primer



MULTIPOINT(10 15, 20 15, 25 5)

00	00	00	00	04	00	00	00	03	01	00	00	00	01	40	24	00	00	00	00	00	00	40	2E
00	00	00	00	00	01	00	00	00	01	40	34	00	00	00	00	00	00	40	2E	00	00	00	00
00	00	01	00	00	00	01	40	39	00	00	00	00	00	00	40	14	00	00	00	00	00	00	

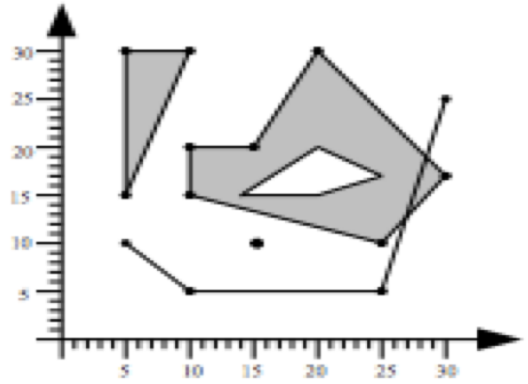
WKT i WKB - Primer



MULTIPOLYGON(((10 5, 10 20, 15 20, 20 30, 30 17, 25 10, 10 5),(15 15, 20 20, 25 17, 20 15, 15 15)),((5 15, 5 30, 10 30, 5 15)))

00	00	00	00	06	00	00	00	02	01	00	00	00	03	00	00	00	02	00	00	00	07	40	14
00	00	00	00	00	00	40	14	00	00	00	00	00	00	40	14	00	00	00	00	00	00	40	34
00	00	00	00	00	00	40	24	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	2E
00	00	00	00	00	00	40	3E	00	00	00	00	00	00	40	39	00	00	00	00	00	00	40	31
00	00	00	00	00	00	40	24	00	00	00	00	00	00	40	34	00	00	00	00	00	00	40	14
00	00	00	00	00	00	40	14	00	00	00	00	00	00	00	00	00	05	40	24	00	00	00	00
00	00	40	2E	00	00	00	00	00	00	40	2E	00	00	00	00	00	00	40	34	00	00	00	00
00	00	40	34	00	00	00	00	00	00	40	39	00	00	00	00	00	00	40	2E	00	00	00	00
00	00	40	2E	00	00	00	00	00	00	40	24	00	00	00	00	00	00	40	2E	00	00	00	00
00	00	01	00	00	00	03	00	00	00	01	00	00	00	04	40	14	00	00	00	00	00	00	40
2E	00	00	00	00	00	00	40	14	00	00	00	00	00	00	40	3E	00	00	00	00	00	00	40
24	00	00	00	00	00	00	40	3E	00	00	00	00	00	00	40	14	00	00	00	00	00	00	40

WKT - Primer



GEOMETRYCOLLECTION(POINT(15 10), LINESTRING(5 10, 10 5, 25 5, 30 25),
MULTIPOLYGON(((10 5, 10 20, 15 20, 20 30, 30 17, 25 10, 10 5),(15 15, 20 20, 25 17, 20 15, 15 15)),((5 15, 5 30, 10 30, 5 15))))

Klasifikacija prostora i model podataka

- Klasifikacija prostora:
  - Velika količina opisa koji omogućuju organizovanje prostora.
  - Topološki model – homomorfne relacije , npr. preklapanje
  - Model Euclidean prostora – razdaljine i pravci u prostoru
  - Grafički model – spajanje, najkraće putanje

- Model prostornih podataka
  - Pravila za identifikaciju objekata i osobina prostora
  - *Objektni* model je pogodan za predstavljanje prostornih entiteta koji imaju neki oblik kao što su jezera, mreža puteva, gradovi i slično.
  - *Field* model se često koristi za prezentovanje kontinualnih i bezobličnih pojava, na primer temperetura oblaka, satelitski snimci, snežne padavine, itd.

Prostorni Odnosi

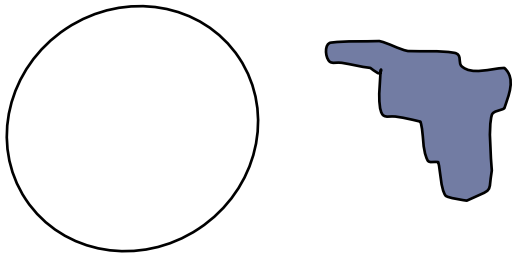
- Tipovi prostora i primer operacija:
  - Topološki - Pored
  - Mrežni - Najkraći put od
  - Direktni - Severno od
  - Euklidov - Udaljenost od

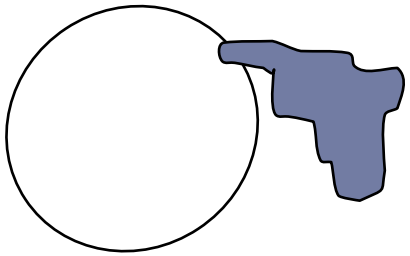
Prostorni odnosi

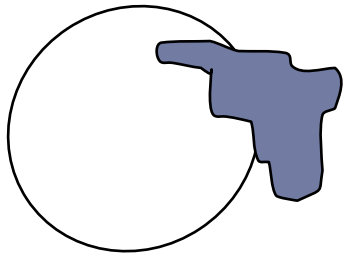
- Bolje razumevanje kompleksne semantike prostornih objekata i operacija na projektantskom nivou
- Jasnost i konzistentnost na korisničkom nivou
- Korak ka standardizaciji

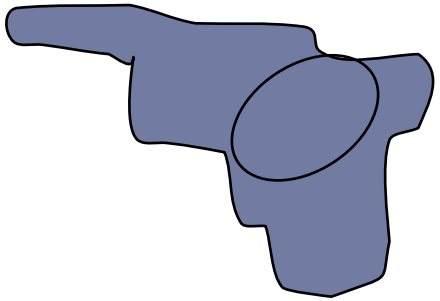
Definisanje prostornih odnosa

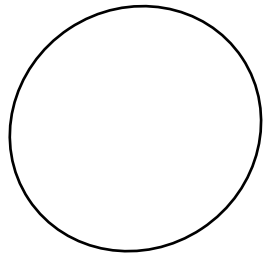
- Pitanje: kako se može precizno definisati prostorni odnos?



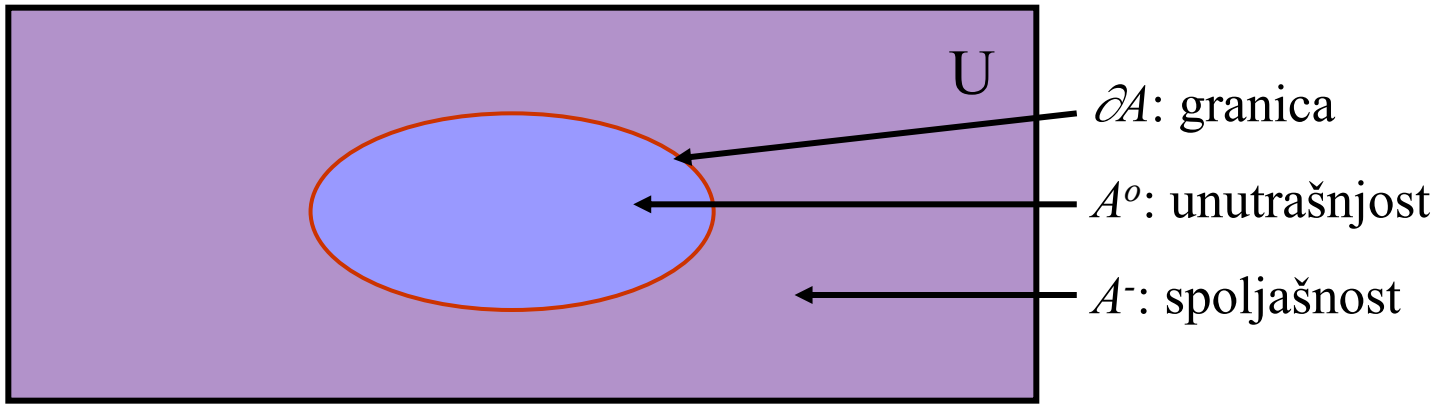








Model 9 – preklapanja



$$\Gamma_9(A,B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

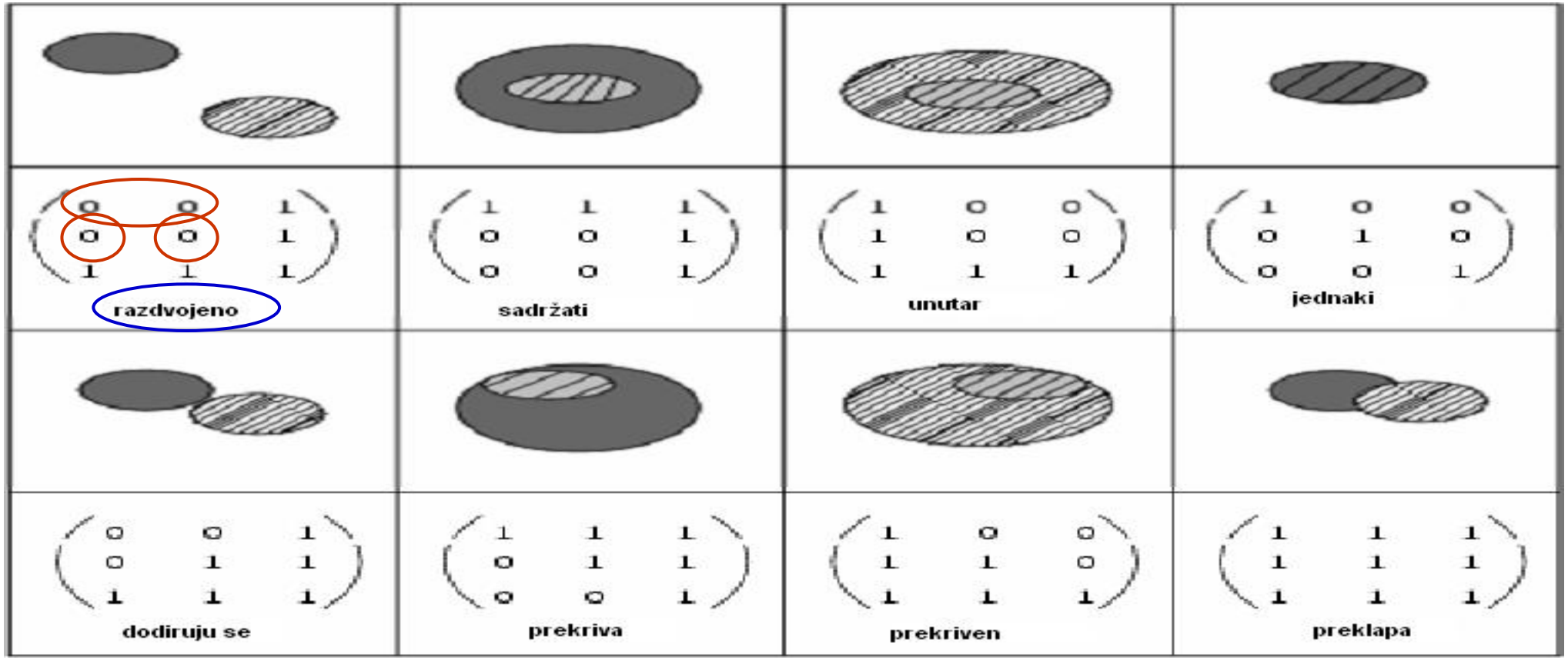
Granice objekata

- Granica se sastoji od tačaka ili linija koje razdvajaju unutrašnjost od spoljašnjosti.
- Granica linije se sastoji od njenih krajnjih tačaka, međutim ako se one poklapaju (tj ako su ista tačka), linija nema granicu.
- Granice multilinije su krajnje tačke svih linija komponenata, međutim ako se one poklapaju, samo krajnje tačke koje se poklapaju neparan broj puta su granice.
- Granica poligona je linija koja opisuje njegov perimetar (obim). Unutrašnjost se sastoji od tačaka koje su u objektu ali ne i na njegovoj granici, a spoljašnjost se sastoji od tačaka koje ne pripadaju objektu.

Devetopresečni (Nine-intersection) model

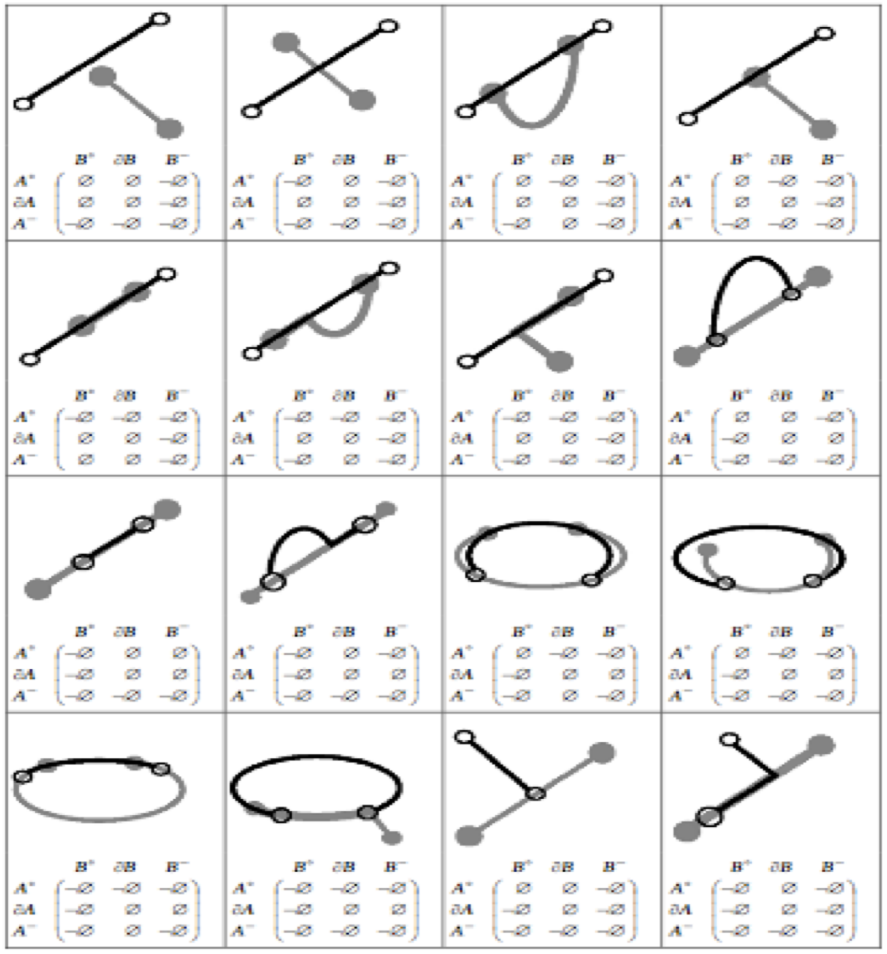
- Objekat A ima tri komponente:
  - granica Ag,
  - unutršnjost Au i
  - spoljašnjost As
- Svaki par objekata ima devet mogućih preseka između njihovih komponenti.
- Parovi komponenti imaju kao rezultat preseka:
  - prazan (0) skup ili
  - neprazan (1) skup.
- Skup preseka između dve geometrije je predstavljen pomoću devetopresečne matrice koja specificira koji parovi komponenti se seku a koji ne.

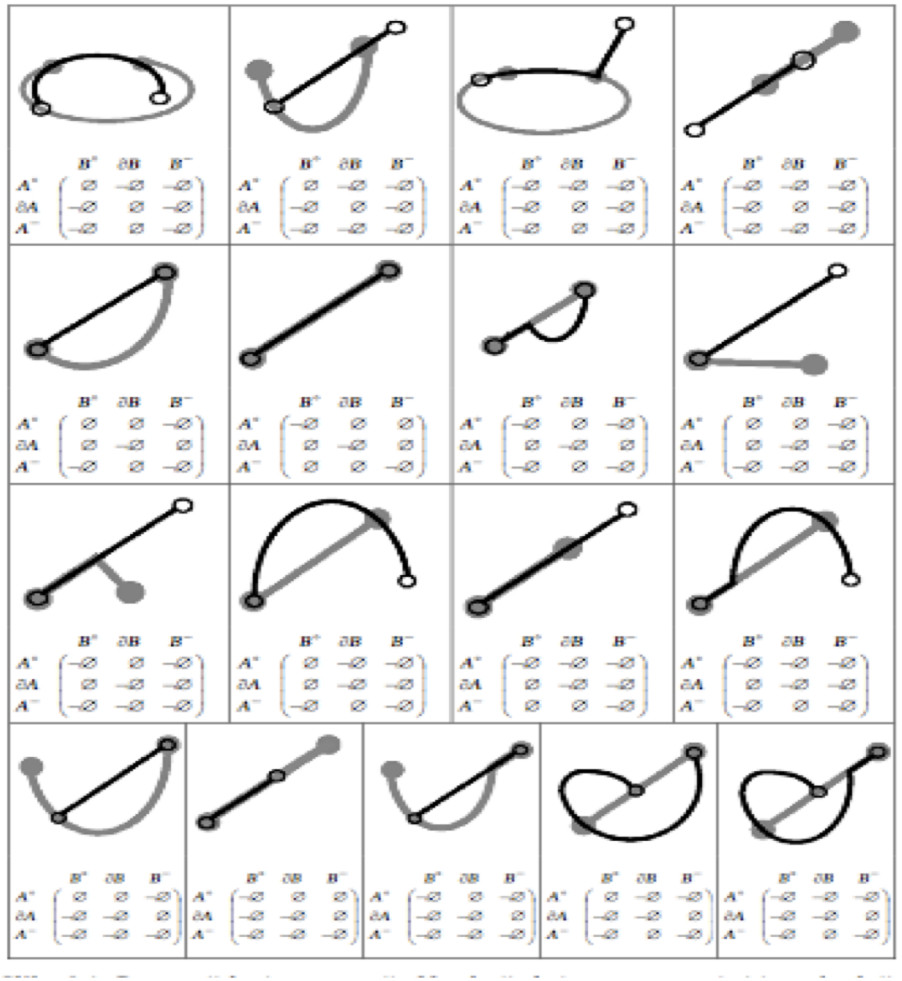
Model 9 – preklapanja



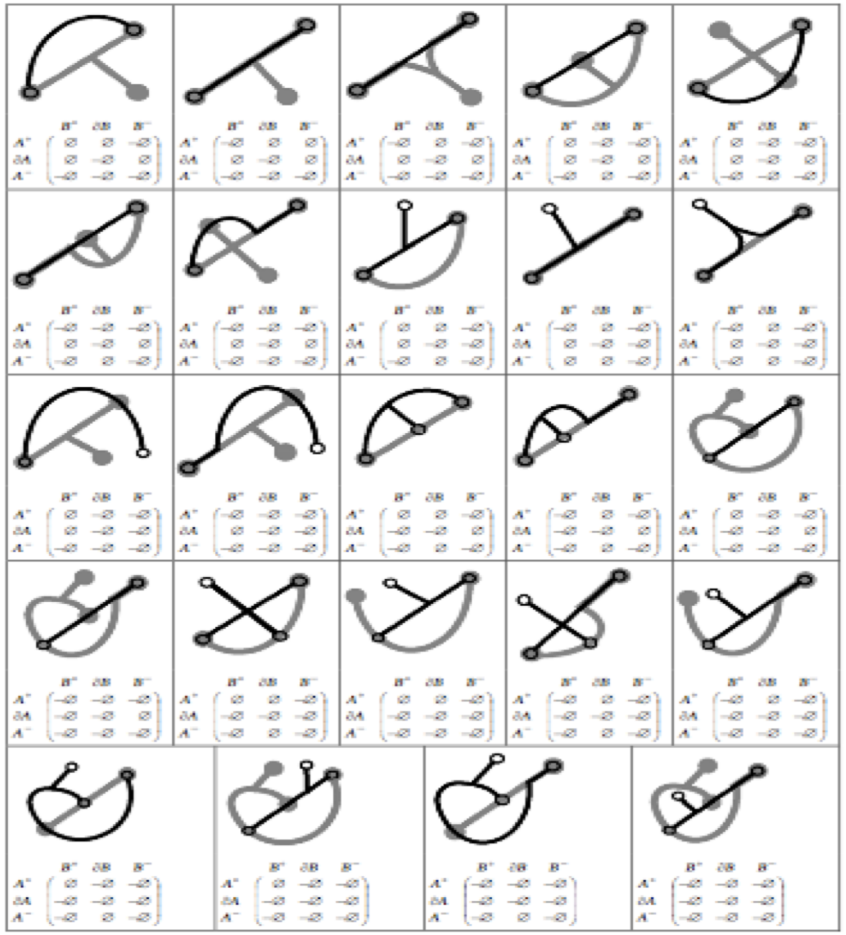
Grupa relacija	Moguće relacije	Relane relacije
Poligon/ Poligon	8	6
Linija / Poligon	19	19
Tačka / Poligon	3	3
Linija / Linija	33	23
Tačka / Linija	3	3
Tačka / Tačka	2	2
		56

33 relacije između dve linije





24 relacije izmedju kompleksih linija



19 relacija regija - linija

Jednostavni Prostorni Upiti

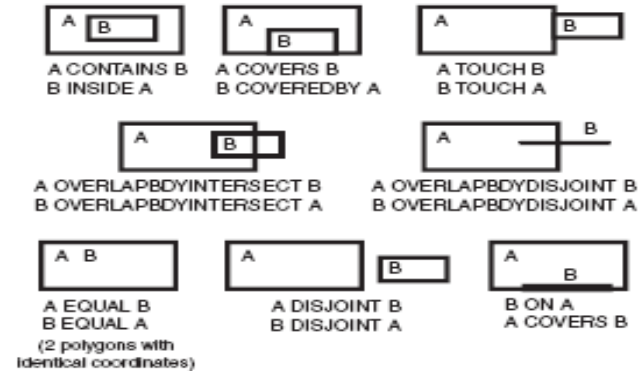
- Upit nad tačkom
  - data je tačka *p*, nađi sve objekte koji sadrže *p*
  - dat je poligon *R*, nađi sve objekte preklopljene sa *R*

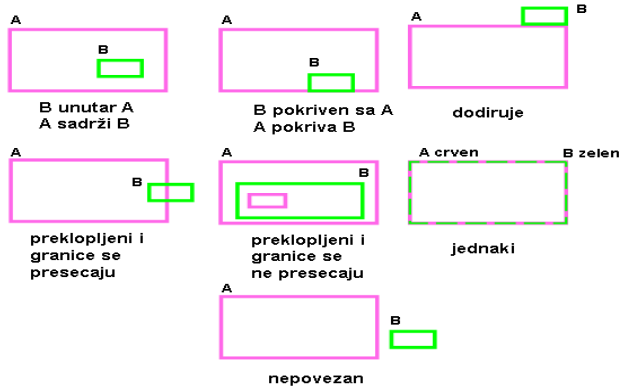
Prostorni upiti

- Fundamentalne operacije
  - selekcija (selection), spajanje (join), projekcije (projection)
- Prostorne funkcije
  - Presek (intersection), preklapanje (overlay), fuzija (fusion), prozoriranje (windowing), isecanje (clipping), centar (centre), granica (boundary)
  - Površina (area), perimeter, rastojanje (distance)
- Drugi prostorni upiti
  - Najbliži sused (nearest neighbour), traženje po sličnosti (similarity search)
- Kompleksni prostorni upiti
  - Višestruki predikati/operacije (multiple predicates/operations), prostorni pod-upiti (spatial sub-queries)

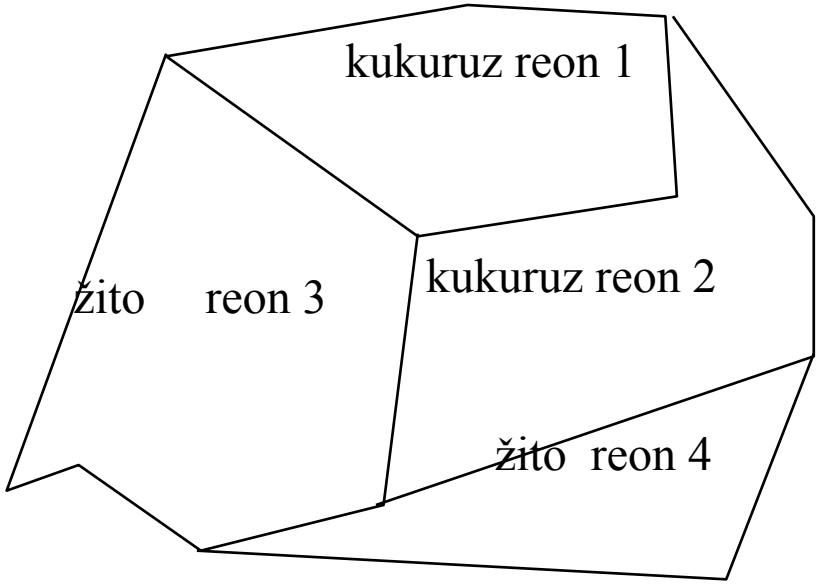
Topološke relacije

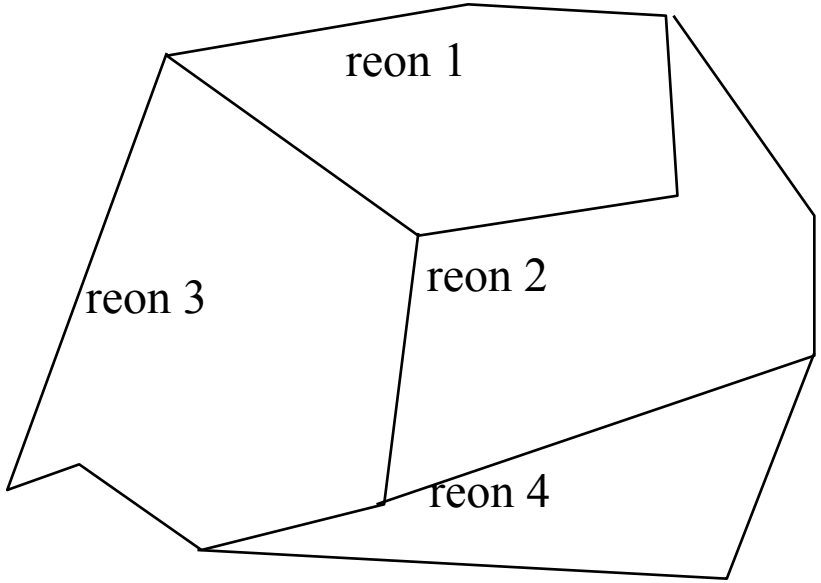
- DISJOINT: nepovezani – granice i unutrašnjost se ne seku.
- TOUCH: dodiruje – granice se seku ali se unutrašnjosti ne seku.
- OVERLAPBDYDISJOINT: preklopljen bez presecanja granica – unutrašnjost jednog objekta preseca granicu i unutrašnjost drugog objekta, ali im se granice ne seku. Primer ove relacije je kada linija počinje izvan poligona a završava unutar poligona.
- OVERLAPBDYINTERSECT: preklopljen sa presecanjem granica – granice i unutrašnjosti dva objekta se seku.
- EQUAL: jednaki – dva objekta imaju istu granicu i unutrašnjost.
- CONTAINS: sadrži – unutrašnjost i granica jednog objekta su kompletno sadržani u unutrašnjosti drugog objekta.
- COVERS: pokriva – unutrašnjost jednog objekta je kompletno sadržana u unutrašnjosti ili granici drugog objekta i njihove granice se seku.
- INSIDE: unutar – suprotno od sadrži (contains). A unutar B implicira B sadrži A.
- COVEREDBY: pokriven sa – suprotno od pokriva (covers). A pokriven sa B implicira B pokriva A.
- ON: na – unutrašnjost i granica jednog objekta je na granici drugog objekta (i drugi objekat pokriva prvi objekat). Primer ove relacija je kada je linija na granici poligona.
- ANYINTERACT: interaguju – objekti nisu nepovezani.



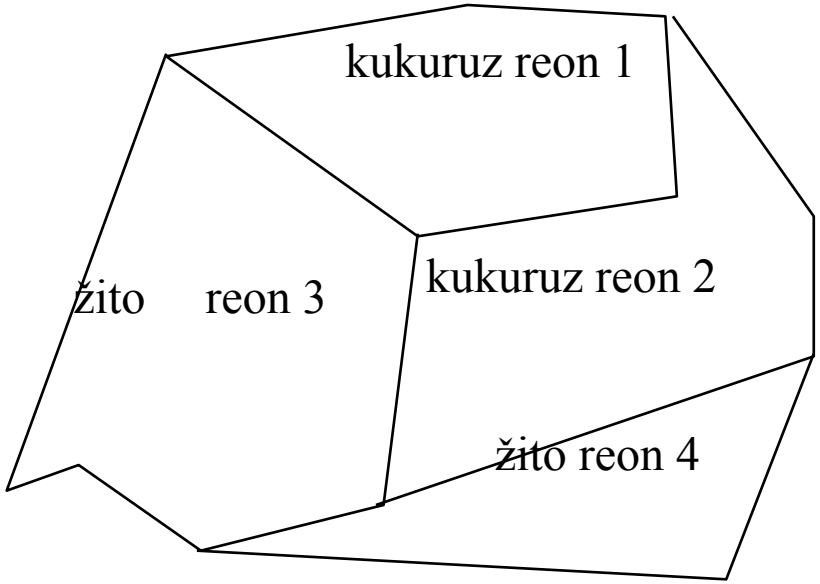


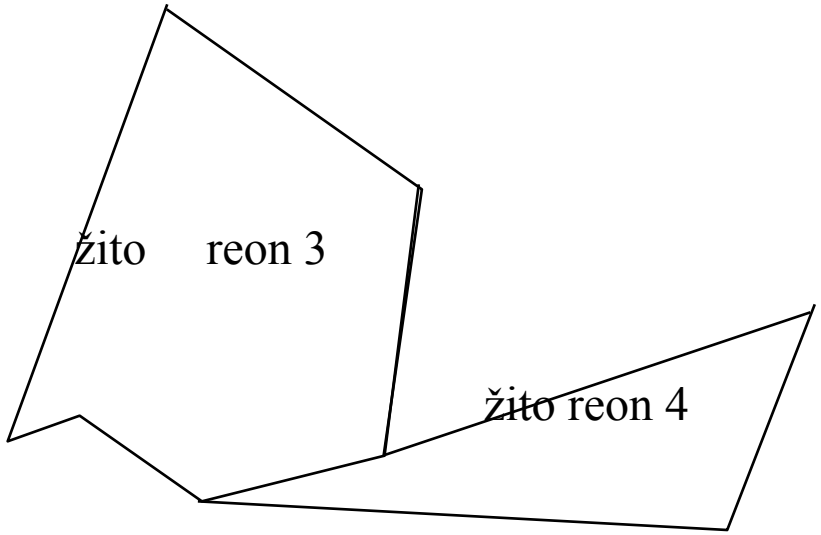
Projekcije (Projection)



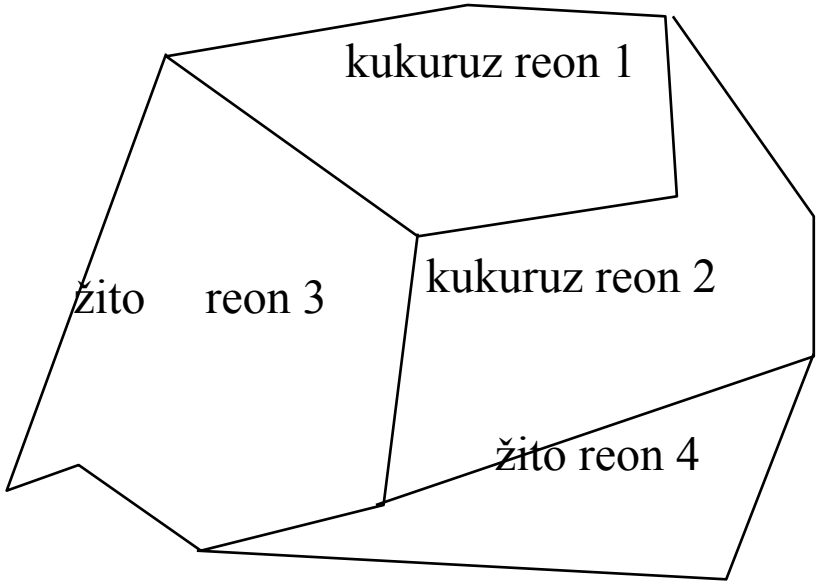


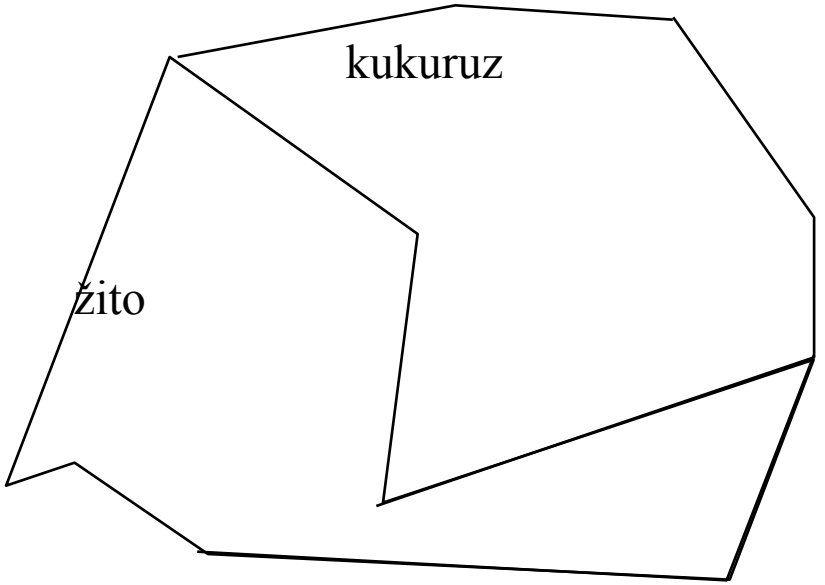
Selekcije (Selection)



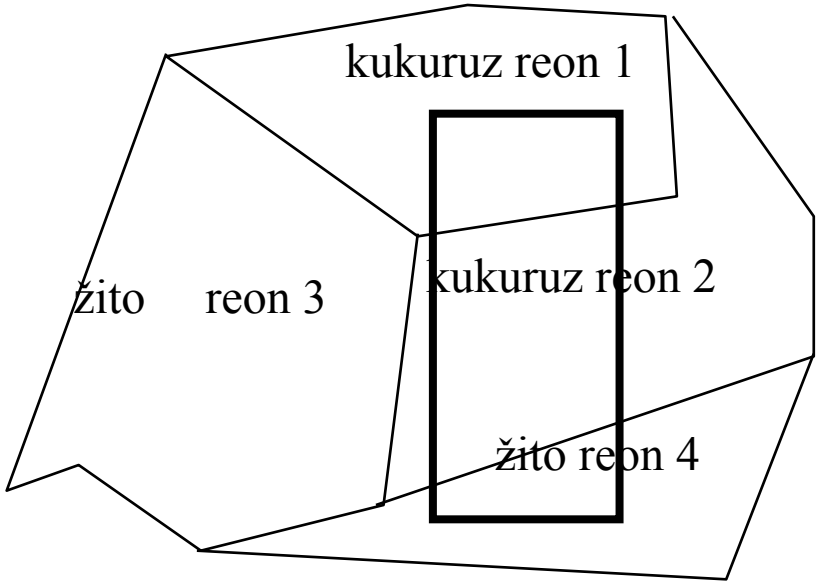


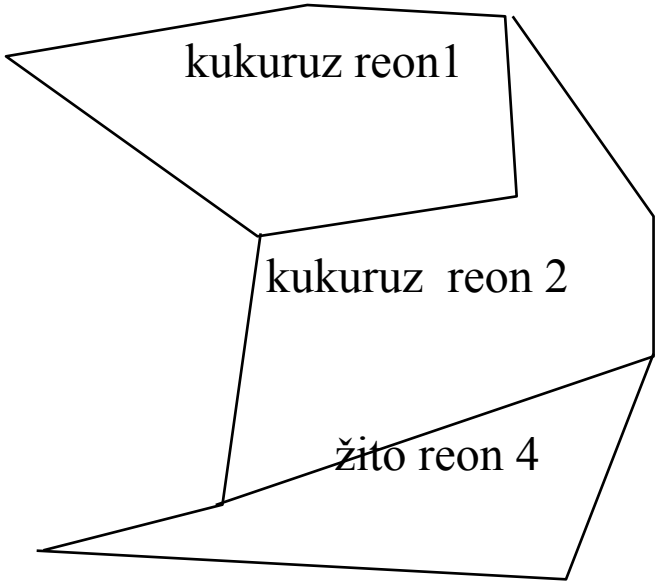
Fuzija/Integracija (Fusion/Amalgamation)



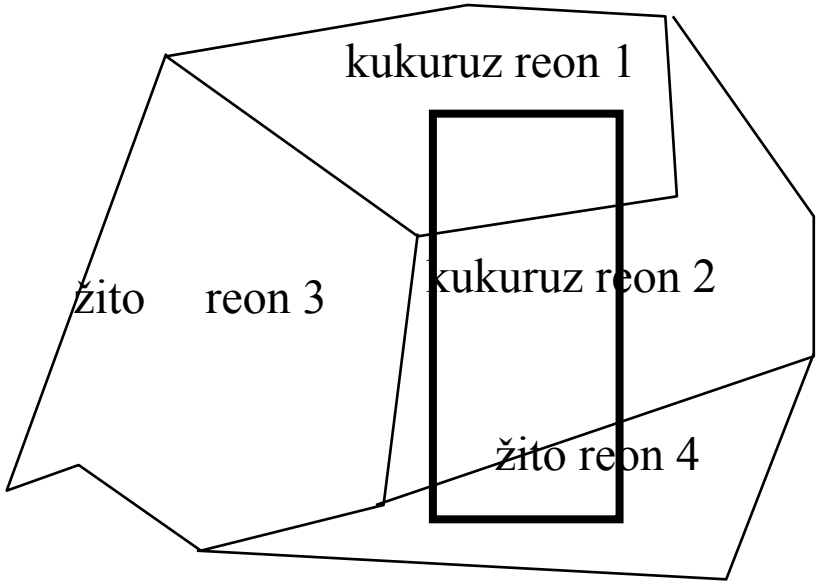


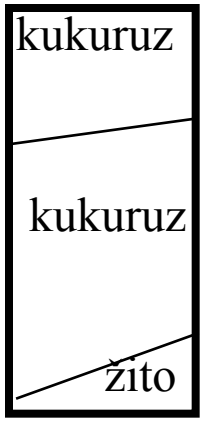
Prozoriranje (Windowing)



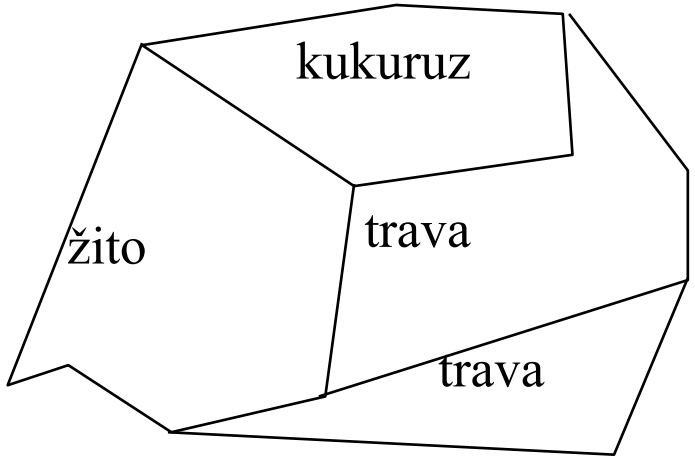


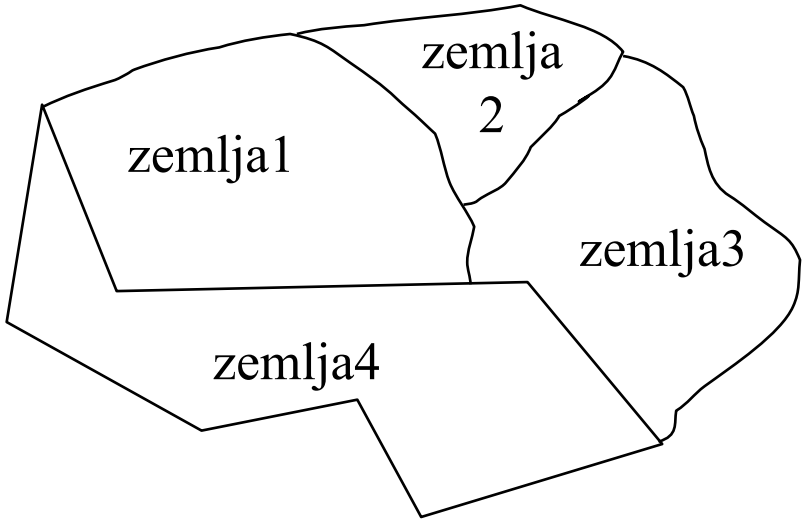
Isecanje (Clipping)

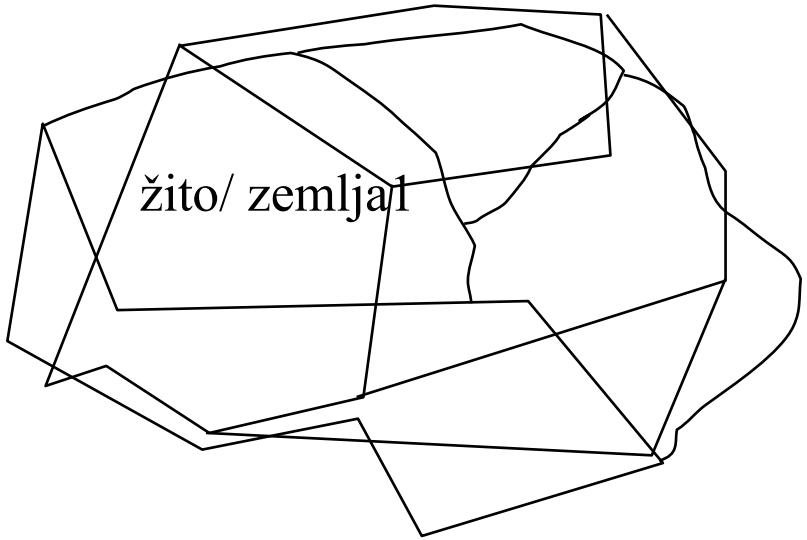




Preklapanje (Overlay)







Prostorni upiti

- Jezik prostornih upita (Spatial query language)
  - Tipovi prostornih podataka, npr. tačka (point), linija (linestring), poligon (polygon), ...
  - Prostorne operacije, npr. preklapanje (overlap), rastojanje (distance), najbliži sused (nearest neighbor), ...
  - Mogu da se pozovu iz prostornih jezika (npr. SQL3) iz podsistema DBMS

```
SELECT      P.name
FROM        Profesor P, Fakultet F
WHERE  P.mesto.Area() > 20 AND
            within (F.lokacija, P.mesto)
```

- Standardi
  - SQL3 (ili SQL 1999) je standard za upitne jezike
  - OGIS je standard za prostorne tipove podataka i operacije nad njima
  - Oba standarda su podržani od strane industrije

Primer

•Primer spajanja kolona u prostornom upitu

```
SELECT      P.name
FROM        Profesor P, Fakultet F
WHERE       P.mesto.Area() > 20 AND
            Within (F.lokacija, P.mesto)
```

• Primer spajanja kolona u ne-prostornom upitu

```
SELECT      P.name
FROM        Profesor P, Fakultet F
WHERE       where P.jmbg = F.jmbg AND
            P.pol = 'ženski'
```

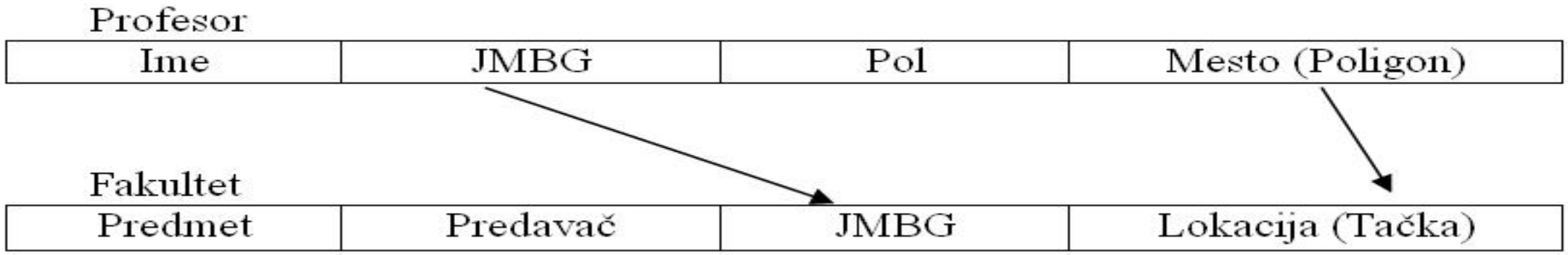


Tabela KATASTARSKA_OPSTINA

KATASTARSKA_OPSTINA		
OID	NOT NULL	NUMBER(10)
SIFRA		CHAR(6)
NAZIV	NOT NULL	VARCHAR2(16)
SLUZBENA_POVRSINA		NUMBER(10)
POVRSINA		NUMBER(10)
GEOMETRIJA	NOT NULL	MDSYS.SDO_GEOMETRY
IZVOR_GEOMETRIJE		VARCHAR2(16)
TRANSACTION_TIME		WMSYS.WM_PERIOD
VALID_TIME		WMSYS.WM_PERIOD
NAPOMENA		VARCHAR2(128)

Tabela PARCELA

PARCELA		
OID	NOT NULL	NUMBER(10)
SIFRA_KO		CHAR(6)
KKULOZAK_PL		NUMBER(6)
BROJ		NUMBER(6)
PODBROJ		NUMBER(6)
PLAN		VARCHAR2(8)
SKICA		VARCHAR2(8)
LIST		VARCHAR2(8)
NAZIV		VARCHAR2(32)
GRADJEVINSKO_ZEMLJISTE	NOT NULL	CHAR(2)
SLUZBENA_POVRSINA		NUMBER(10)
POVRSINA		NUMBER(10)
AREA		NUMBER(12,2)
BLOK		NUMBER(8)
KATASTAR_NEKRETNINA	NOT NULL	CHAR(2)
GEOMETRIJA	NOT NULL	MDSYS.SDO_GEOMETRY
IZVOR_GEOMETRIJE		VARCHAR2(16)
TRANSACTION_TIME		WMSYS.WM_PERIOD
VALID_TIME		WMSYS.WM_PERIOD
NAPOMENA		VARCHAR2(128)

Tabela ZGRADA

ZGRADA		
OID	NOT NULL	NUMBER(10)
SIFRA_KO		CHAR(6)
BROJ_PARCELE		NUMBER(6)
PODBROJ_PARCELE		NUMBER(6)
BROJ		VARCHAR2(6)
VRSTA_ZGRADE		VARCHAR2(24)
SEKTOR_VLASNISTVA		VARCHAR2(12)
GODINA_IZGRADNJE		DATE
OSNOVA_IZGRADNJE		CHAR(2)
VRSTA_OBJEKTA		CHAR(2)
BROJ_SPRATOVA		CHAR(2)
KORISNA_POVRSINA		NUMBER(8)
GEOMETRIJA	NOT NULL	MDSYS.SDO_GEOMETRY
IZVOR_GEOMETRIJE		VARCHAR2(16)
TRANSACTION_TIME		WMSYS.WM_PERIOD
VALID_TIME		WMSYS.WM_PERIOD
NAPOMENA		VARCHAR2(32)

Kreiranje prostronih indeksa nad SDO_GEOMETRY kolonom

```
CREATE INDEX kat_opstina_idx ON katastarska_opstina (geometrija)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

CREATE INDEX parcela_idx ON parcela (geometrija)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;

CREATE INDEX zgrada_idx ON zgrada (geometrija)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

SDO_FILTER

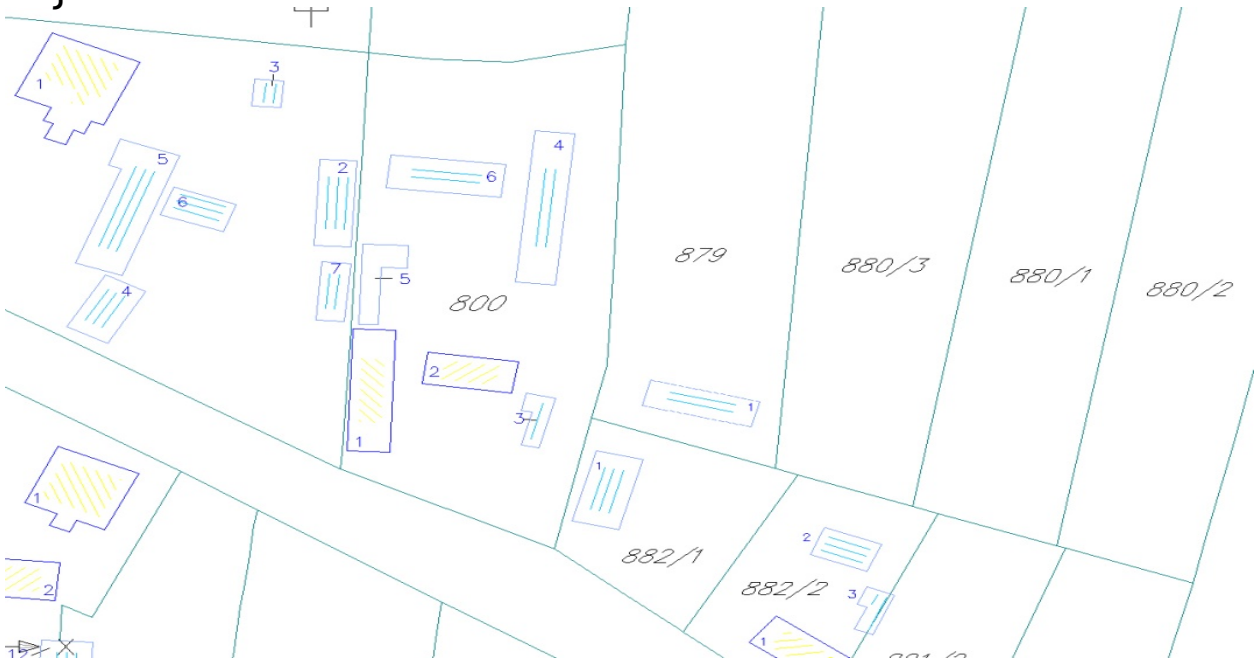
Koje zgrade možda interaguju sa parcelom broj 800?

```
SELECT p.broj, z.broj, z.broj_parcele

FROM parcela p, zgrada z

WHERE SDO_FILTER(p.geometrija, z.geometrija)='TRUE' AND
    p.broj=800;
```

BROJ	BROJ	BROJ_PARCELE
800	1	882
800	2	797
800	7	797
800	3	800
800	4	800
800	5	800
800	6	800
800	1	800
800	2	800



SDO_RELATE [1/5]

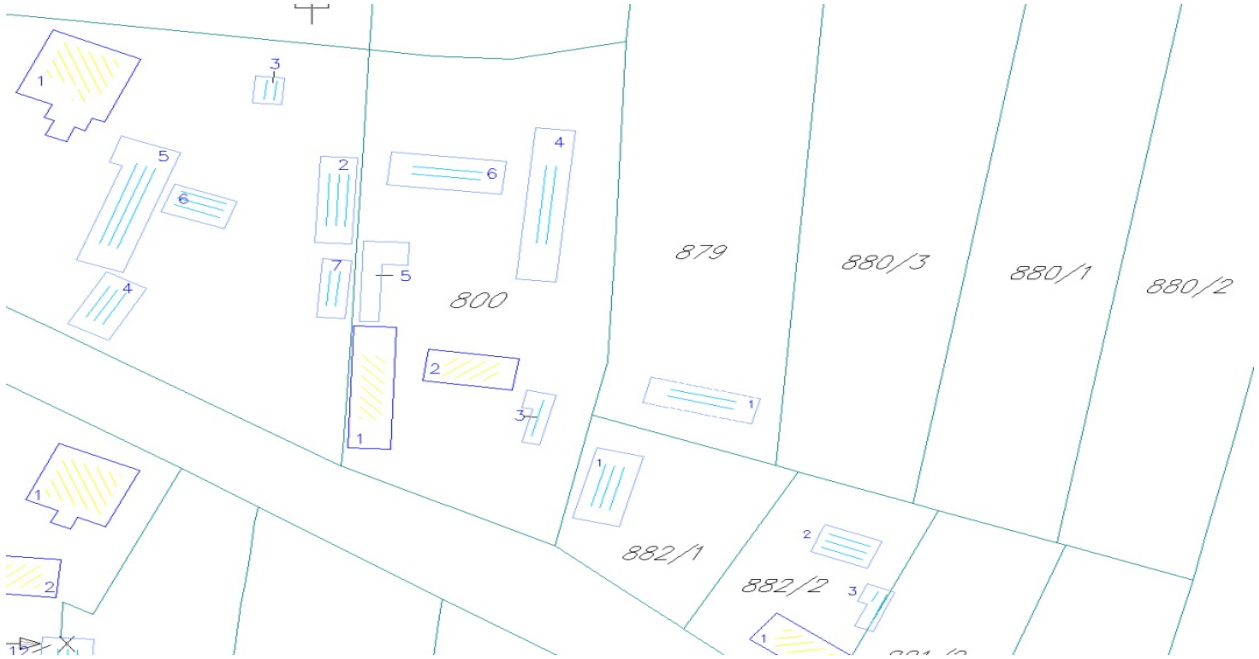
Koje zgrade interaguju sa parcelom broj 800?

```
SELECT p.broj, z.broj, z.broj_parcele

FROM parcela p, zgrada z

WHERE SDO_RELATE(p.geometrija, z.geometrija,
    'mask=anyinteract')='TRUE' AND p.broj=800;
```

BROJ	BROJ	BROJ_PARCELE
800	3	800
800	4	800
800	5	800
800	6	800
800	1	800
800	2	800



SDO_RELATE [2/5]

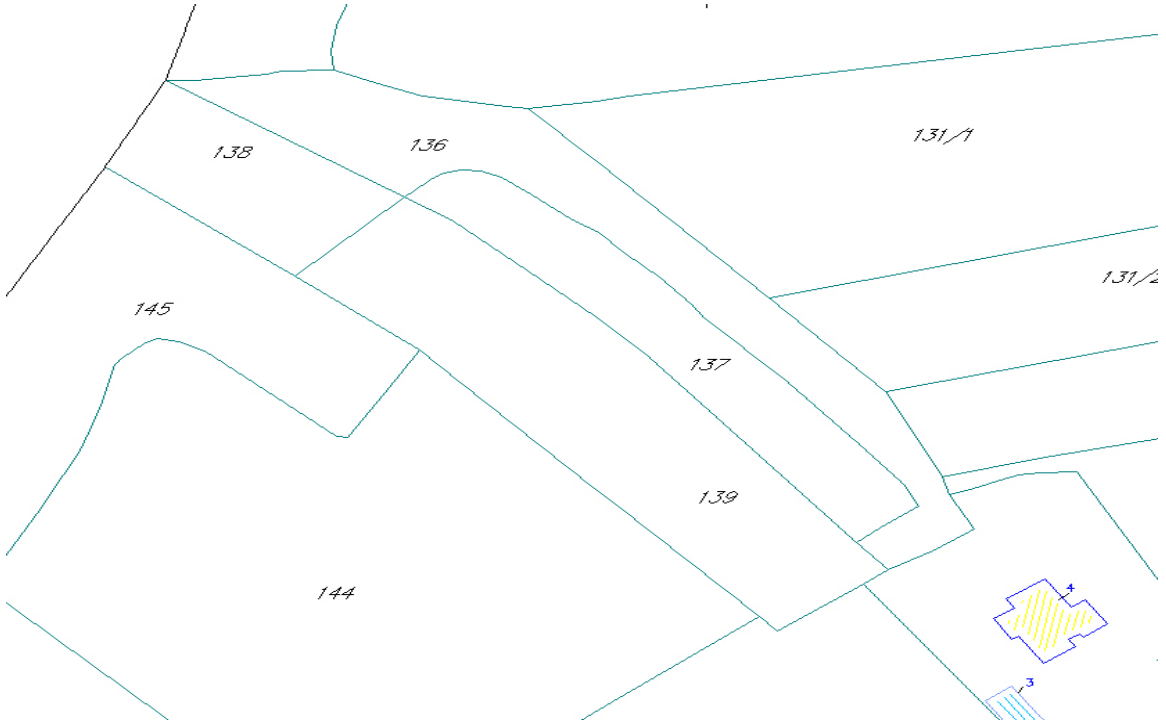
Sa kojim parcelama, parcela broj 137, stoji u prostornom odnosu “dodiruje”?

```
SELECT p.broj
FROM parcela p137, parcela p
WHERE p137.broj = 137
AND   SDO_RELATE(p.geometrija, p137.geometrija, 'mask=touch') =
    'TRUE';

ILI

SELECT p.broj
FROM parcela p137, parcela p
WHERE p137.broj = 137
AND   SDO_TOUCH(p.geometrija, p137.geometrija) = 'TRUE';
```

```
BROJ
----------
138
136
139
```



SDO_RELATE [3/5]

Sa kojim zgradama parcela broj 800 stoji u prostornom odnosu “sadrži”?

```
SELECT p.broj, z.broj
FROM parcela p, zgrada z
WHERE SDO_RELATE(p.geometrija, z.geometrija, 'mask=contains')='TRUE'
    AND p.broj=800;
```

ILI

```
SELECT p.broj, z.broj
FROM parcela p, zgrada z
WHERE SDO_CONTAINS(p.geometrija, z.geometrija)='TRUE' AND
    p.broj=800;
```

BROJ	BROJ
800	3
800	4
800	5
800	6
800	1
800	2

SDO_RELATE [4/5]

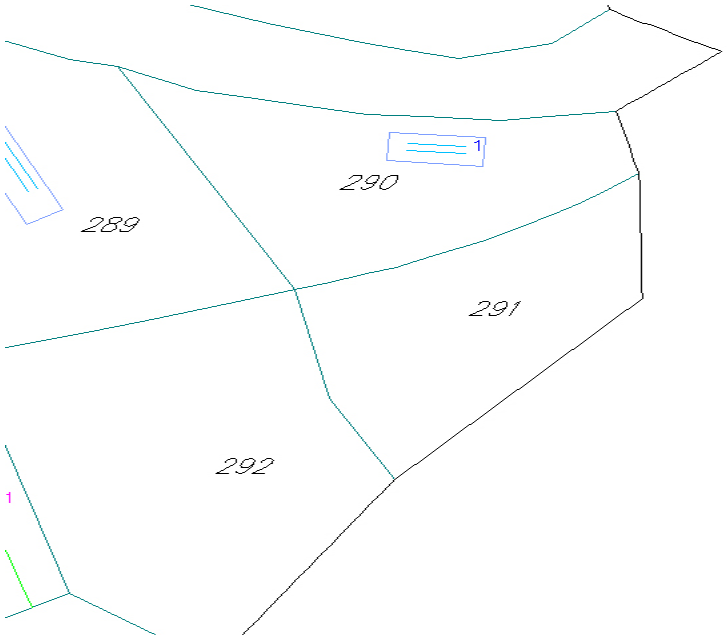
Da li katastarska opština Popovi1 stoji u prostornom odnosu “pokriva” sa parcelom broj 291?

```
SELECT SDO_RELATE(k.geometrija, p.geometrija, 'mask=covers')
FROM katastarska_opstina k, parcela p
WHERE k.naziv = 'Popovi1' AND p.broj = 291;

ILI

SELECT SDO_COVERS(k.geometrija, p.geometrija)
FROM katastarska_opstina k, parcela p
WHERE k.naziv = 'Popovi1' AND p.broj = 291;
```

```
SDO_COVERS(K.GEOMETRIJA,P.GEOMETRIJA)
-------------------------------------
TRUE
```



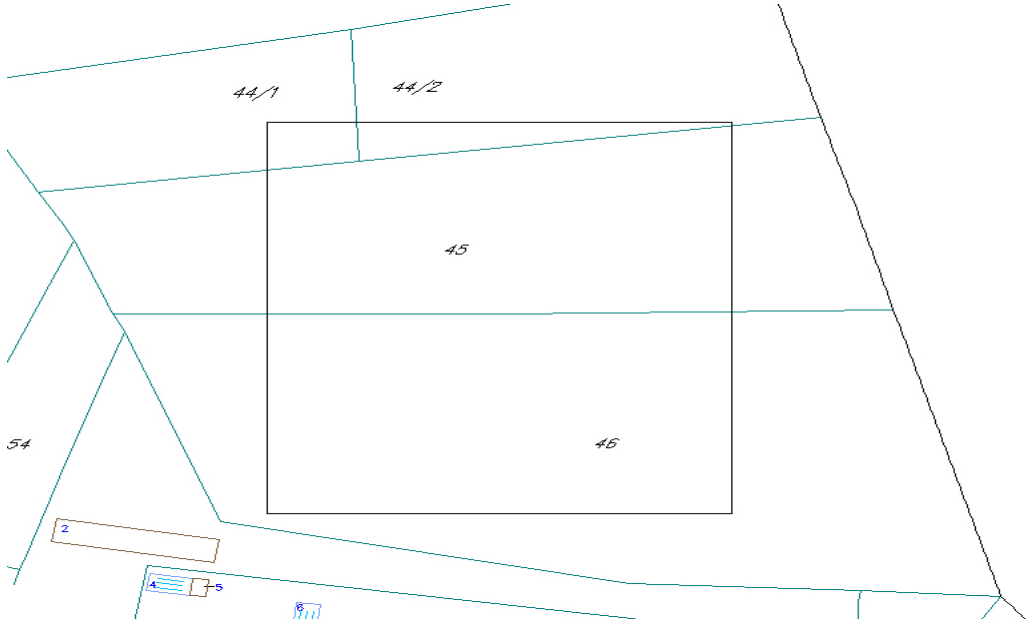
SDO_RELATE [5/5]

Koje parcele preklapa zadati pravougaonik?

```
SELECT p.broj, p.podbroj
FROM parcela p
WHERE SDO_RELATE(p.geometrija,
SDO_GEOMETRY(2003, 31276, NULL,
SDO_ELEM_INFO_ARRAY(1,1003,3),
SDO_ORDINATE_ARRAY(6602283.97,4959931.97,
    6602359.48,4960016.78)),
'mask= overlapbdyintersect') = 'TRUE';

ILI

SELECT p.broj, p.podbroj
FROM parcela p
WHERE SDO_OVERLAPBDYINTERSECT(p.geometrija,
SDO_GEOMETRY(2003, 31276, NULL,
SDO_ELEM_INFO_ARRAY(1,1003,3),
SDO_ORDINATE_ARRAY(6602283.97,4959931.97,
    6602359.48,4960016.78))
) = 'TRUE';
```



BROJ	PODBROJ
44	1
45	
46	
44	2

SDO _WITHIN _DISTANCE [1/2]

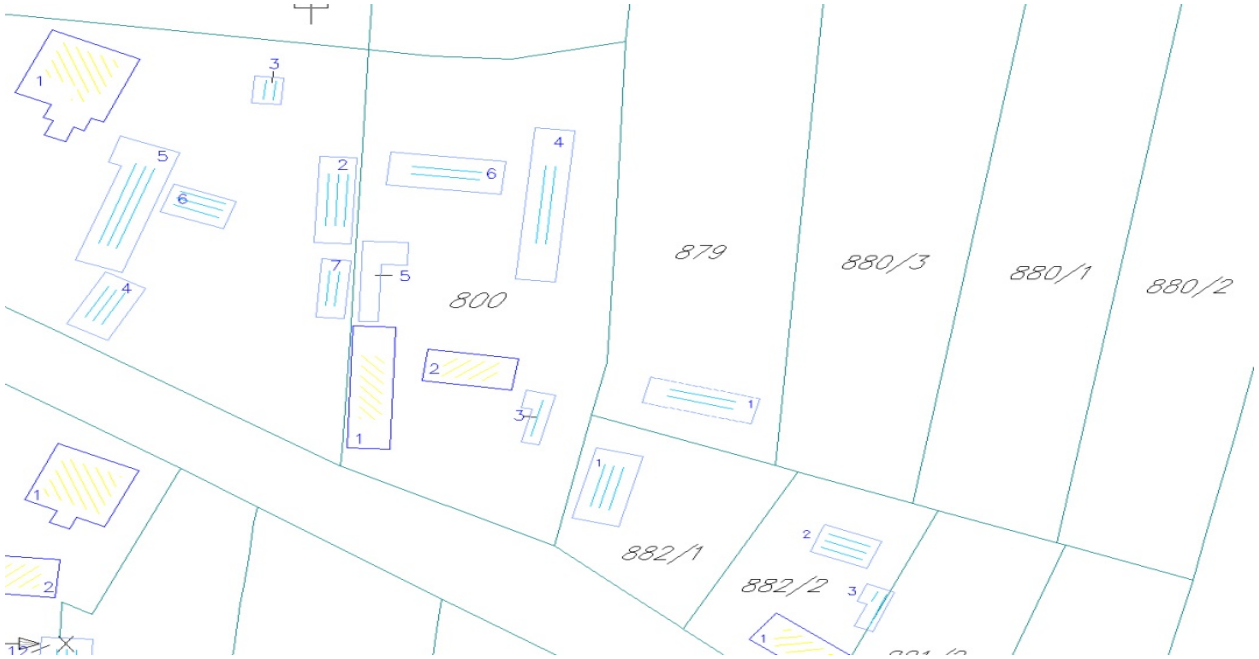
Koje zgrade se nalaze unutar razdaljine od 10m od zgrade broj 2 na parceli 800?

```
SELECT z.broj, z.broj_parcele

FROM zgrada z, zgrada z2

WHERE z2.broj_parcele=800 AND z2.broj=2 AND

SDO_WITHIN_DISTANCE(z.geometrija, z2.geometrija,
     'distance=10' ) = 'TRUE';
```



BROJ	BROJ_PARCELE
1	800
2	800
5	800
7	797
3	800

SDO _WITHIN _DISTANCE [2/2]

Koje zgrade se nalaze unutar razdaljine od 5m od zgrade broj 2 na parceli 800?

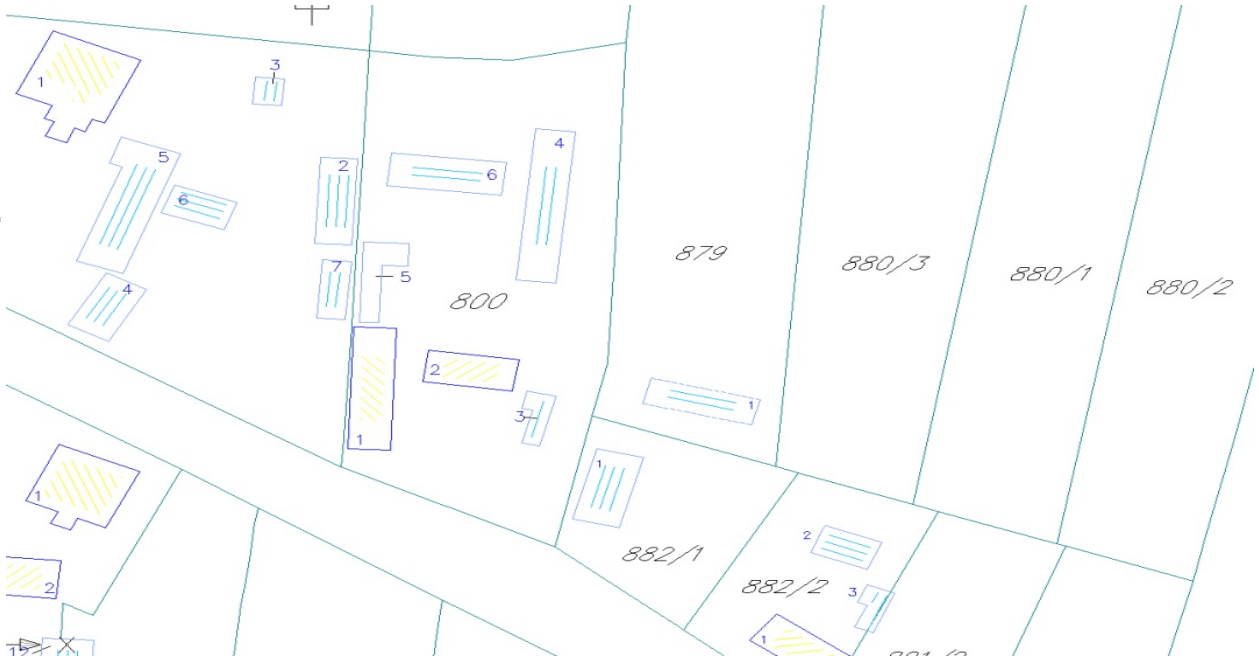
```
SELECT z.broj, z.broj_parcele

FROM zgrada z, zgrada z2

WHERE z2.broj_parcele=800 AND z2.broj=2 AND

SDO_WITHIN_DISTANCE(z.geometrija, z2.geometrija, ‘distance=5’
    'TRUE';
```

BROJ	BROJ_PARCELE
1	800
2	800
3	800



SDO_NN [1/4]

Kojih pet zgrada su najbliže zgradi broj 2 na parceli 800?

Upotreba ključne reči sdo_num_res.

```
SELECT /*+ INDEX(parcela parcela_idx) */ z.broj, z.broj_parcele

FROM zgrada z, zgrada z2

WHERE z2.broj_parcele=800 AND z2.broj=2 AND

SDO_NN(z.geometrija, z2.geometrija, 'sdo_num_res=5') = 'TRUE';
```

BROJ	BROJ_PARCELE
1	800
2	800
7	797
3	800
5	800

SDO_NN [2/4]

Kojih pet zgrada su najbliže zgradi broj 2 na parceli 800 i kolika je njihova razdaljina?

Upotreba ključne reči sdo_num_res.

```
SELECT /*+ INDEX(parcela parcela_idx) */  z.broj, z.broj_parcele,
    SDO_NN_DISTANCE(1)

FROM zgrada z, zgrada z2

WHERE z2.broj_parcele=800 AND z2.broj=2 AND

SDO_NN(z.geometrija, z2.geometrija, 'sdo_num_res=5', 1) =
    'TRUE';
```

BROJ	BROJ_PARCELE	SDO_NN_DISTANCE(1)
1	800	2.93010956
2	800	0
7	797	9.69618482
3	800	1.50709999
5	800	6.40824469

SDO_NN [3/4]

Kojih pet zgrada su najbliže zgradi broj 2 na parceli 800?

Upotreba ključne reči sdo_batch_size.

```
SELECT /*+ INDEX(parcela parcela_idx) */ z.broj, z.broj_parcele

FROM zgrada z, zgrada z2

WHERE z2.broj_parcele=800 AND z2.broj=2 AND

SDO_NN(z.geometrija, z2.geometrija, 'sdo_batch_size=5') =
    'TRUE' AND rownum<=5;
```

BROJ	BROJ_PARCELE
2	800
3	800
1	800
5	800
7	797

SDO_NN [4/4]

Kojih pet zgrada su najbliže zgradi broj 2 na parceli 800 i kolika je njihova razdaljina?

Upotreba ključne reči sdo_batch_size.

```
SELECT /*+ INDEX(parcela parcela_idx) */  z.broj, z.broj_parcele,
    SDO_NN_DISTANCE(1)

FROM zgrada z, zgrada z2

WHERE z2.broj_parcele=800 AND z2.broj=2 AND

SDO_NN(z.geometrija, z2.geometrija, 'sdo_batch_size=5', 1) = 'TRUE
    AND rownum<=5;
```

BROJ	BROJ_PARCELE	SDO_NN_DISTANCE(1)
2	800	0
3	800	1.50709999
1	800	2.93010956
5	800	6.40824469
7	797	9.69618482

SDO_JOIN

Koje zgrade pripadaju kojoj parceli?

```
SELECT /*+ ordered */ a.broj, b.broj, b.broj_parcele

FROM TABLE(SDO_JOIN('parcela', 'geometrija', 'zgrada', 'geometrija',
     'mask=COVERS+CONTAINS')) c,

parcela a,

zgrada b

WHERE c.rowid1 = a.rowid AND c.rowid2 = b.rowid

ORDER BY a.broj;
```

...

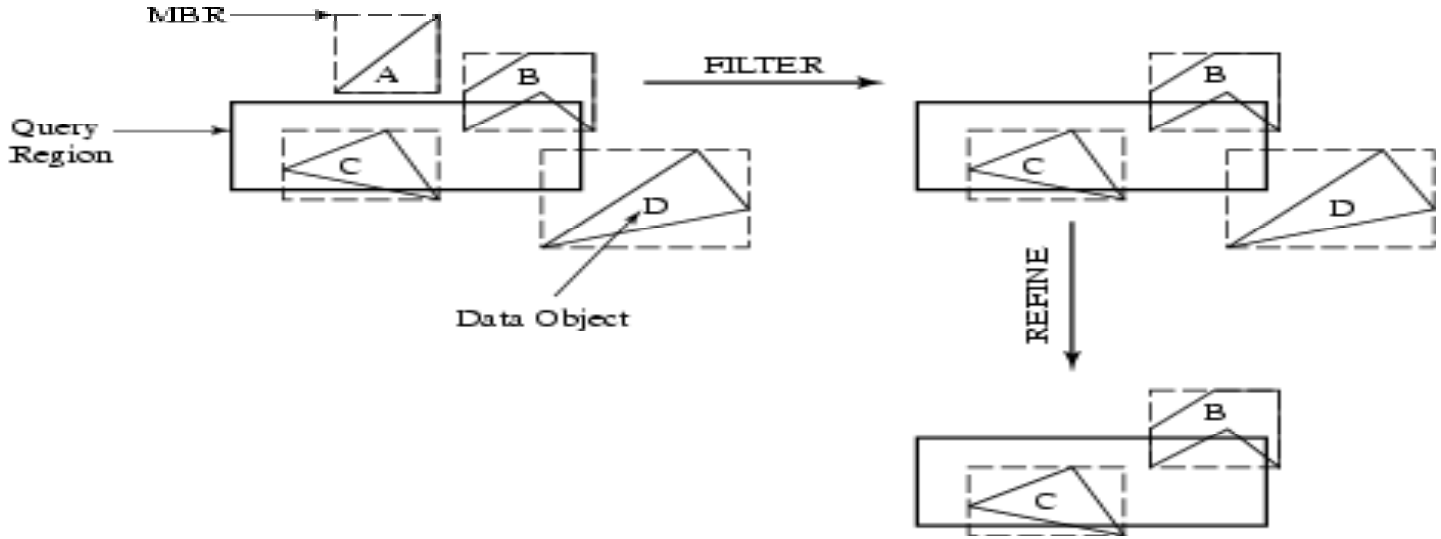
BROJ	BROJ	BROJ_PARCELE
793	4	793
793	3	793
797	1	797
797	7	797
797	6	797
797	5	797
797	4	797
797	3	797
797	2	797
800	5	800
800	2	800

BROJ	BROJ	BROJ_PARCELE
800	1	800
800	4	800
800	3	800
800	6	800
803	5	803
803	6	803
803	7	803
803	8	803
803	1	803
803	3	803
803	4	803

...

Procesiranje Upita

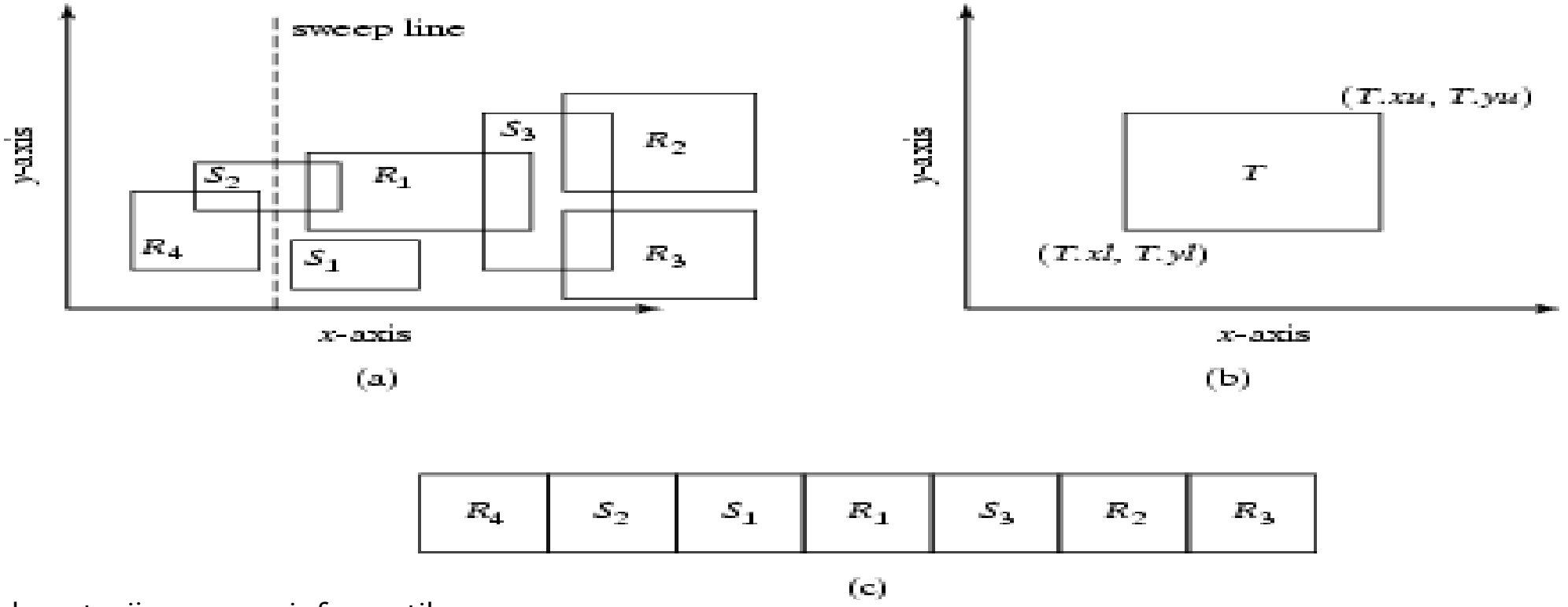
• Postoje efikasni algoritmi koji daju odgovore na prostorne upite
• Algoritam se bazira na filtriranju i prolazi kroz dva koraka
• U prvom koraku objekat koji se nalazi pod upitom se predstavlja minimalnim pravougaonicima (MBR). Ovaj korak se naziva filter korak, zbog toga što mnogi kandidati koji odgovaraju na upit budu eliminisani
• Drugi korak je procesiranje rezultata prvog filter koraka, ali sada sa originalnim i tačnim geometrijama.



Procesiranje i Spajanje Upita

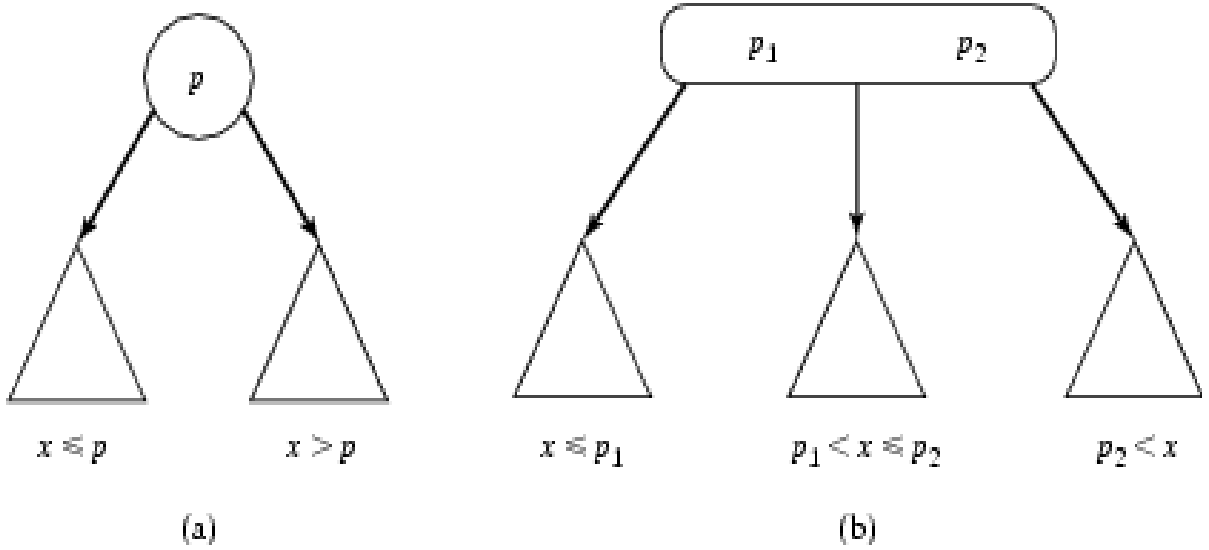
•Primer – Definisanje parova praovugaonika pod presekom

- (a):Dva skupa pravougaonika R i S, (b): Pravougaonika sa markiranim tačkama u suprotnim uglovima, (c): Pravougaonici sortirani po najmanjoj vrednosti X koordinate
- Sweep filtriranje identifikuje 5 parova od 12 za korak čišćenja

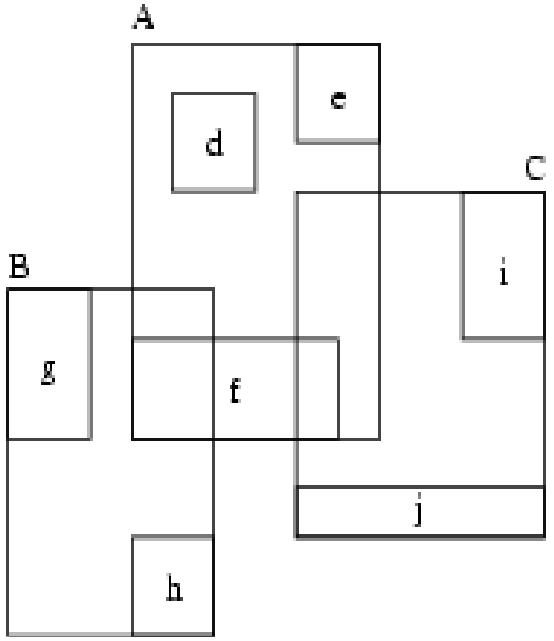


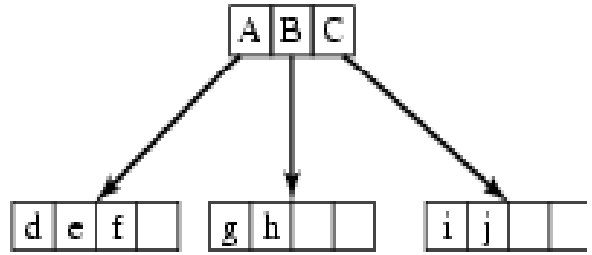
Prostorno Indeksiranje: Pretraga Struktura Podataka

- Mogućnosti prostornog indeksiranja:
  - B-tree je hijerarhijska kolekcija linearnih opsega, npr. brojevi
  - B-tree indeksiranje se uspešno koristi u pretrazi nad tradicionalnim tipovima podataka
  - B-tree mogu da se koriste i nad prostornim podacima
  - R-tree ima bolje performanse pretrage
  - R-tree je hijerarhijska kolekcija pravougaonika



B-tree





R- tree

Optimizacija Upita

•Optimizacija Upita
- Operacije nad prostornim podacima mogu da koriste različite strategije
- "Cena" svake strategije zavisi od različitih parametara
- Optimizacija upita predstavlja proces:
  - •redosled operacija u upitu
  - •izbor efikasne strategije za svaku operaciju posebno
  - •bazira se na detaljima o datim podacima

•Primer upita:

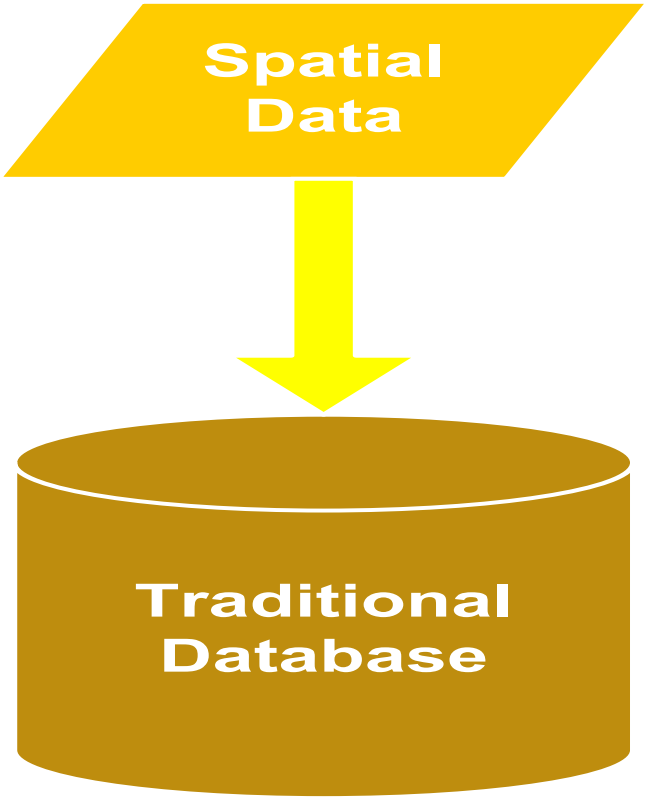
```
SELECT    P.name
FROM      Profesor P, Fakultet F
WHERE     where P.jmbg = F.jmbg AND
          P.pol = 'ženski'
```

•Primer odluke optimizacije
- •Prvo se procesira (P.pol = 'ženski') pa onda (P.jmbg = F.jmbg)
- •Indeksiranje se ne koristi za procesiranje (P.jmbg = F.jmbg)

Primarni i sekundarni filter

- **Primarni filter** omogućuje brzu selekciju kandidata datog upita koje zatim prosleđuje sekundarnom filteru. Primarni filter upoređuje aproksimacije geometrija da bi smanjio kompleksnost izračunavanja i smatra se manje zahtevnim filterom. S obzirom da primarni filter upoređuje aproksimacije geometrija, on vraća nadskup tačnog rezultujućeg skupa.

- **Sekundarni filter** primenjuje tačna izračunavanja na geometrijama koje su rezultat primarnog filtera. Sekundarni filter daje tačan odgovor na prostorni upit. On je zahtevan u pogledu izračunavanja, ali se primenjuje samo na rezultat primarnog filtera, a ne na celokupan skup podataka.


Spatial
Data
Traditional
Database

IBM

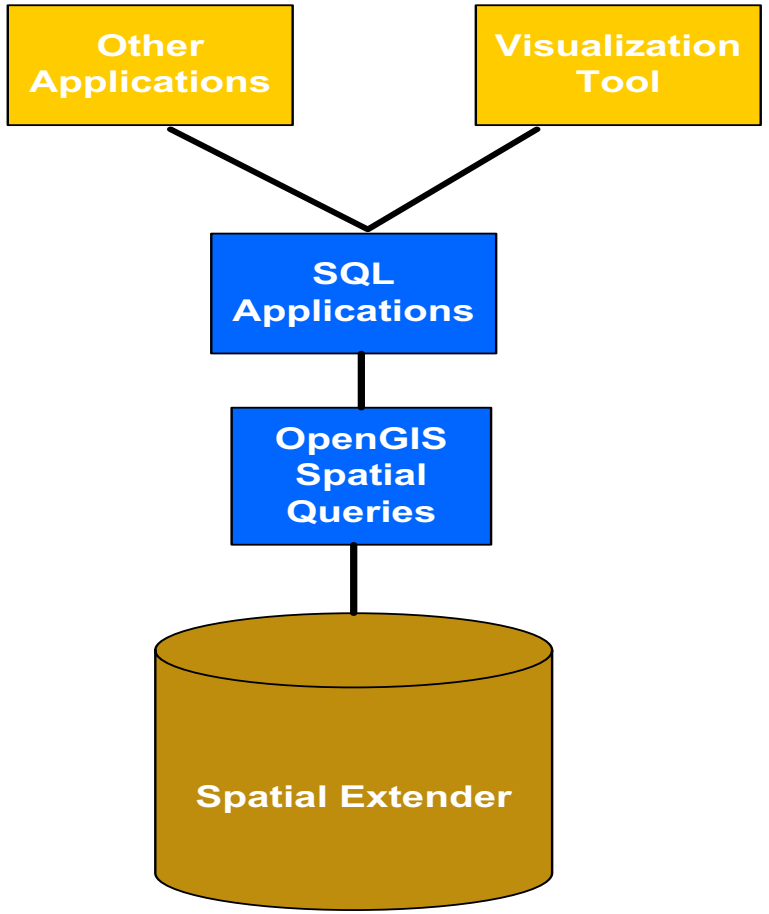
PostGIS

ORACLE

ESRI

ArcSDE

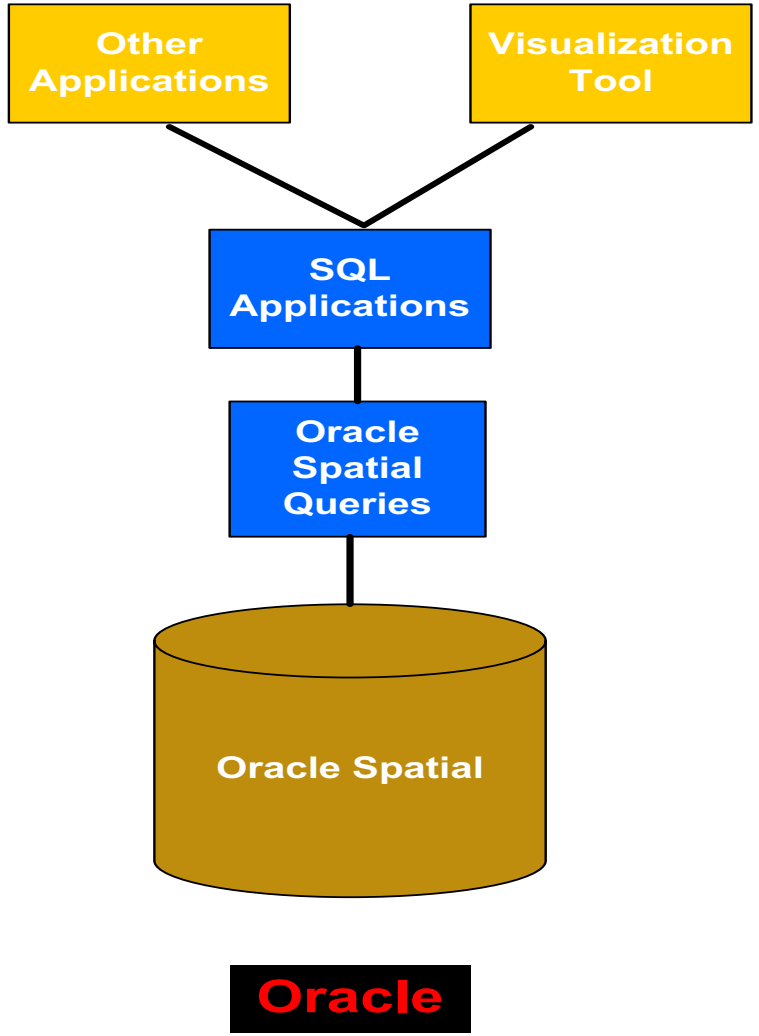
- **IBM DB2 Spatial Extender**
  - Spatial information, in formats such as point, lines, and polygons, can be analyzed, generated, stored, accessed, and queried within IBM DB2.
  - The combination of spatial data with other, more traditional type data, adds another layer of intelligence to the database.



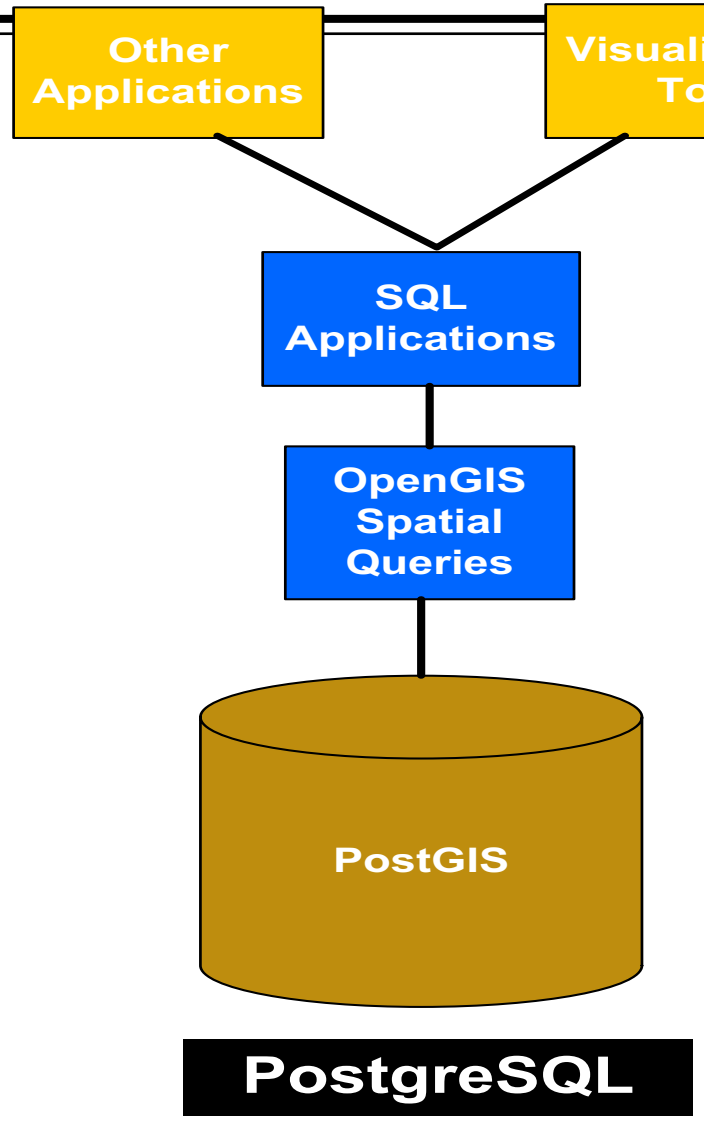
Domain Spatial Database Product

- Oracle Spatial

  - Oracle Spatial, an option for Oracle Database 10g Enterprise Edition, includes advanced spatial capabilities to support GIS applications, location-based services, and enterprise spatial information systems.
  - Oracle Spatial extends the core location features included in every Oracle database with Oracle Locator.



- **PostGIS with PostgreSQL Database**
  - adds geographic object support to PostgreSQL.
  - is an open source, object-relational database that runs from the server
  - enables PostgreSQL to store, relate, join, query, etc… with spatial data.
  - is capable of running on Windows, Linux, and Unix operating systems.

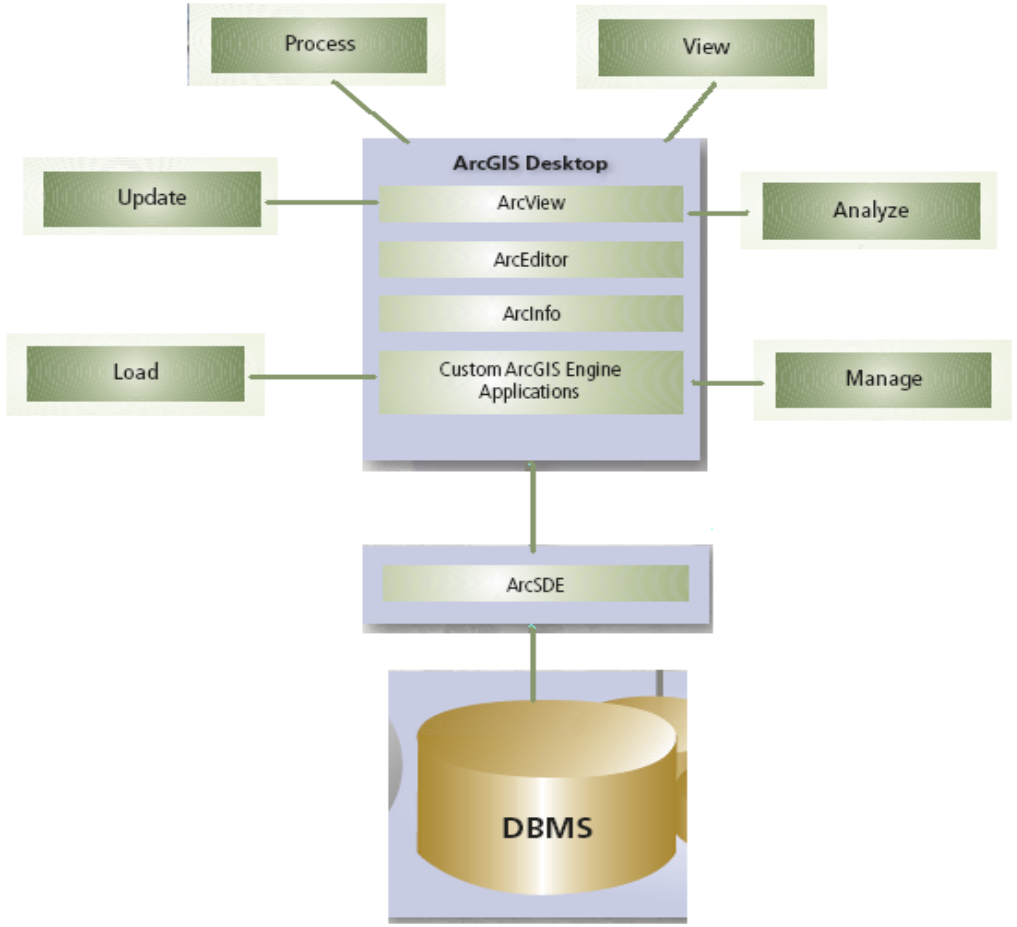


Domain Spatial Database Product

- ArcSDE

- ArcSDE is middleware – server software product used to augment the range of data types to include spatial data managed in a traditional RDBMS.

- ArcSDE serves data to the ArcGIS Desktop products (ArcView, ArcEditor, and ArcInfo) or to the Internet. Users perform any process that ArcGIS Desktop supports, such as load, update, view, analyze, manage



Comparison - Cost

Spatial Database Product	*Cost*
IBM DB2 spatial extender	For earlier versions of DB2, cost to add SpatialExtender $9,250 Now included in DB2v8.1 & higher: IBM DB2 Personal Edition $461 IBM DB2 cost varies with number of licenses: Express Edition $4,874 to Enterprise Server Edition $33,125
ArcSDE	Start from $10,000 for commercial version
Oracle Spatial	Enterprise Edition: • Named user license: $800 • Processor license: $40,000 Oracle Spatial: • Named user license: $200 • Processor license: $10,000
PostgreSQL / PostGIS	Refractions Research under the GNU General Public License developed PostGIS for open source, free to the public distribution. It and the PostgreSQL database are completely open-source and can be easily downloaded over the internet for free.

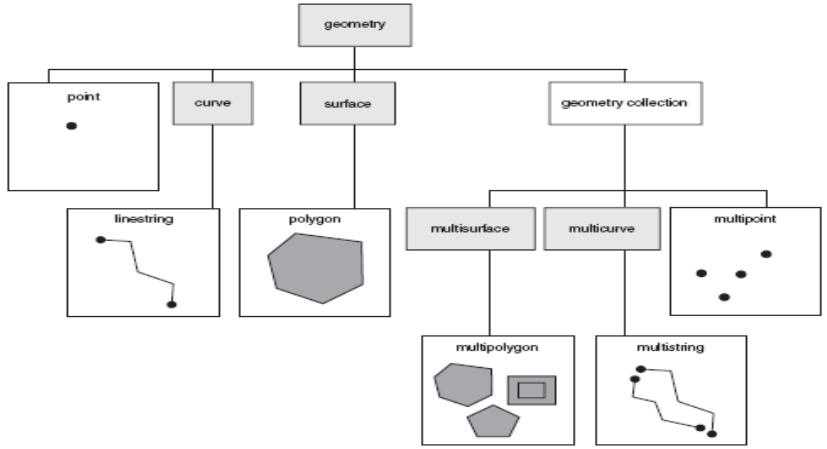
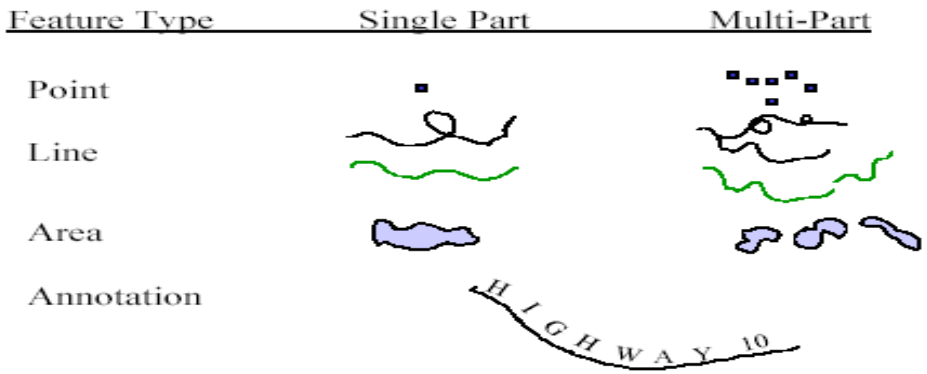
Comparison – Standards

Spatial Database Product	*Standards*
IBM DB2 spatial extender	Conforms to the ISO SQL/MM Spatial Standard and the OpenGIS Consortium's (OGC's) Simple Feature Specification for SQL
ArcSDE	ArcSDE supports OpenGIS simple features and SQL statements.
Oracle Spatial	Oracle spatial implements the OpenGIS Consortium Simple Features guidelines. Oracle is also committed to supporting the new OGC Geographic Markup Language (GML) as well as Open Location Service interfaces.
PostgreSQL / PostGIS	Version 1.0 of PostGIS is currently under validation of the Open GIS Consortium “Simple Features Specification for SQL” which defines standard GIS object types, the functions required to manipulate them, and a set of metadata tables.

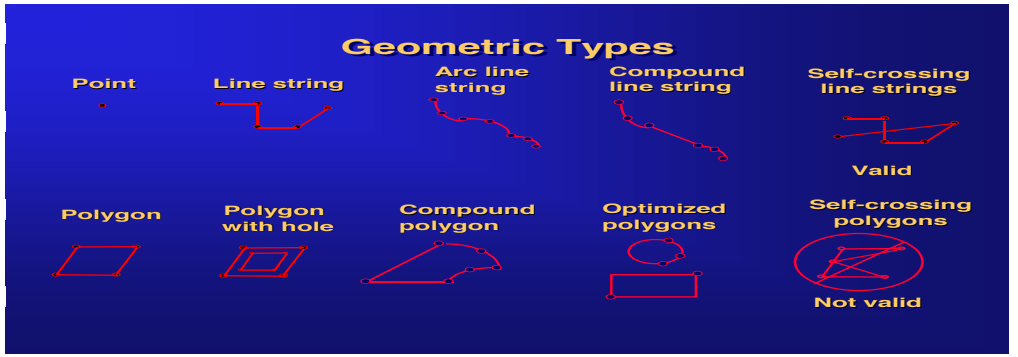
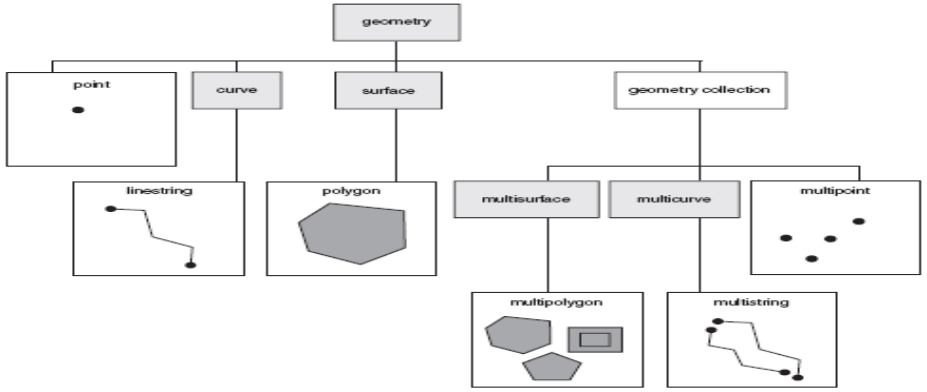
Comparison - Application

Spatial Database Product	*GIS Application*
IBM DB2 Spatial Extender	In 2001 IBM and ESRI partnered and jointly developed the IBM DB2 Spatial Extender, which incorporated geographic or spatial information with business and other data already residing in an IBM Universal Database. Spatial data may be manipulated within the database by utilizing spatial extender functions to return GIS information in tabular format. A separate visualization tool is needed to see a graphic representation such as a map.
ArcSDE	ArcSDE is a server software product used to access massively large multiuser geographic databases stored in relational database management systems (RDBMSs). It is an integrated part of ArcGIS and a core element of any enterprise GIS solution. Its primary role is to act as the GIS gateway to spatial data stored in a RDBMS. ArcSDE provides a suite of services that enhance data management performance, extend the range of data types that can be stored in a RDBMS, enable schema portability between RDBMSs, and offer configuration flexibility.
Oracle Spatial	Oracle Spatial is an extension to Oracle9i that provides geometry storage, indexing, and spatial search functions. When used with Oracle9i Enterprise Edition, Oracle Spatial provides a standards-based data management solution for the delivery of GIS, Internet mapping, and mobile location-based services. It serves as a technology platform for managing an enterprisewide GIS or location-enabled e-business applications.
PostgreSQL / PostGIS	The software itself is a limited standalone GIS, but there are other open source applications that can be downloaded to enhance it's GIS capabilities.

Comparison

Spatial Database Product	*Spatial Data Type*	*Graphic*
IBM DB2 spatial extender	• Points • Lines • Polygons Ability to "subtype" these basic types	
ArcSDE	Support all the ESRI geometry types	

Comparison

Spatial Database Product	*Spatial Data Type*	*Graphic*
Oracle Spatial	SDO_GEOMETRY – can support three geometric primitive types: • Points. • Line Strings (can be linear, curved or both) • Polygons	
PostgreSQL / PostGIS	• Point, Multipoint • Line, Multiline • Polygon,,Multipolygon, • Geometrycollections	

Comparison

Spatial Database Product	*Query*	*Functions*
IBM DB2 spatial extender	• SQL access to spatial data • SQL querying of spatial data and/or joining of spatial data with conventional database or business data	80 built-in spatial data functions 8 geometry type constructor functions 12 spatial data comparison functions 20 geometric conversion functions 40 geometric calculator functions Examples: ST_LineString, ST_Intersects, ST_AsShape, ST_Area
ArcSDE	The spatial query in ArcSDE and RDBMS client is implemented in the form of SQL statements. Requests for data occur when actions like zooming in/out, querying rows in table, and displaying tabular or spatial data are performed. These types of requests are translated into SQL statements and passed to the RDBMS for processing.	All functions just depend on ArcGIS software, such as ArcMap

Comparison

Spatial Database Product	*Query*	*Functions*
Oracle Spatial	Oracle Spatial uses a two-tier process for querying spatial data. • The first part of the query selects candidate rows based only on their spatial index. • The second step of the query looks at each candidate and determines if it meets the query exactly. This two-tier approach allows for quick querying by leveraging the spatial indexes.	Over 400 spatial functions, Oracle Spatial includes functions for length/area calculations, buffer, centroid, convexhull, geometry intersection, union. Such as: 3 SDO_GEOMETRY object type method 17spatial operators 21 Geometry Function 5 Spatial Aggregate Functions 3 Coordinate System Transformation Functions 40 Linear Referencing Functions 6 Migration Procedures 11 Tuning Functions and Procedures 2 Utility Functions
PostgreSQL / PostGIS	Spatial queries are performed the same way they are constructed in any other database query.	Management Functions AddGeometryColumn, DropGeometryColumn, SetSRID Relate Functions Distance, Equals, Disjoint, Intersects, Touches, Crosses, Within, Overlaps, Contains, Intersects, Relate Processing Functions Centroid, Area, Length, PointOnSurface, Boundary, Buffer, ConvexHull, Intersection, Difference, GeomUnion, Envelope, IsSimple, IsClosed, IsRing, NumPoints, ExteriorRing, NumInteriorRings, EndPoint, StartPoint, GeometryType, X, Y, & Z.

Comparison

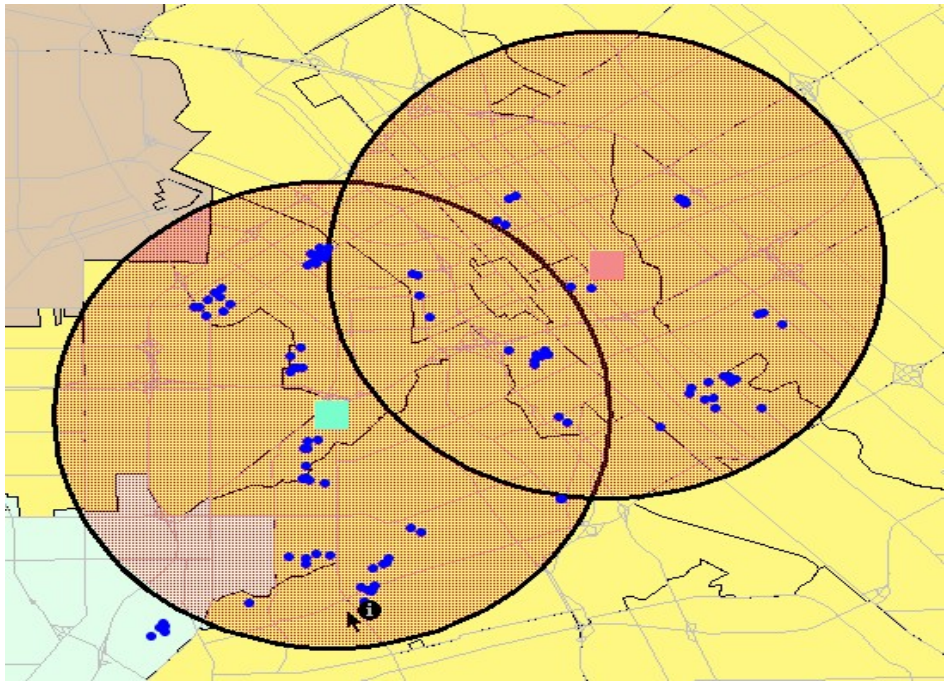
Spatial Database Product	*Data Loading*
IBM DB2 spatial extender	Spatial data may be imported into and exported from Spatial Extender through data exchange files. Two types are supported: • shapefiles and ESRI SDE transfer files. • Also supported are industry standard data types: Well Known Binary (OGC WKB) andd Well Known Text (OGC WKT)
ArcSDE	• Raster data: ArcSDE supports the following formats: ERDAS IMAGINE, ERDAS(.lan & .gis), ERDAS Raw, GIF, BIL/BIP/BSQ, BMP, MrSID compressed images, JPEG, ADRG, PNG, ER Mapper, CIB, CADRG. • Vector data: shape file, AutoCAD, coverage, tab file for MapInfo etc.
Oracle Spatial	• Writing SQL statement to create table & load data into the table directly. • Convert Shape to SDO 1. Using *shp2sdo* to create load assistant files 2. Using *sqlplus* to create table 3. Using *sqlldr* to load the actual data into the table Other data formats should be converted into shapefile before loading.
PostgreSQL / PostGIS	Loading data can be done two ways. • The first way is using the loader, which converts ESRI Shape files into SQL suitable for loading in PostGIS/PostgreSQL. This is similar to importing data into an ESRI Geodatabase. • The other way to get data into the database are converting the data to a text representation, then use traditional SQL data loading statements.

Comparison

Spatial Database Product	*Compatibility with GIS Applications*	*Server Capability*
IBM DB2 spatial extender	Supports visualization tools such as ESRI's ArcView GIS, ArcExplorer, ArcInfo, ArcView BusinessAnalyst as well as other vendors such as MapInfo. To view graphic or map representation, a visualization tool must be used.	If deployed within DB2 edition with server capabilities. But IBM recommends an architecture utilitizing ESRI's ArcSDE for distributed computing.
ArcSDE	Since ArcSDE if a part of ESRI GIS software fimaly, it supports all ESRI GIS software. At the same time, it support AutoCAD by Autodesk.	Serves spatial data to ArcGIS Desktop (ArcReader, ArcView, ArcEditor, and ArcInfo), to Internet clients through ArcIMS, and to applications developed with ArcGIS Engine and ArcGIS Server.
Oracle Spatial	Oracle Spatial is an option for Oracle Enterprise Edition that provides advanced spatial features to support high-end GIS and LBS solutions.	Integration with Oracle9*i* Application Server
PostgreSQL / PostGIS	Available GIS applications that can run PostGIS are uDig (User-friendly Desktop Internet GIS) and MapServer. Both are internet-web applications that can create, acquire, view, and modify spatial data.	PostgreSQL, an open source, object-relational database that is used in conjunction with PostGIS acts as the server. It is capable of running on Windows, Linux, and Unix operating systems.

Selection Summary

- **Comparison Criteria:**
  - Cost
  - Standards met
  - GIS application
  - Spatial data types supported
  - Queries from the database
  - Spatial functions
  - Data loading
  - Compatibility with other GIS applications
  - Server capability
  - Ability to be stand alone or integration with database



Retail selection based on customers and proximity to competitors.