Przetwarzanie i analiza danych przestrzennych, GIS

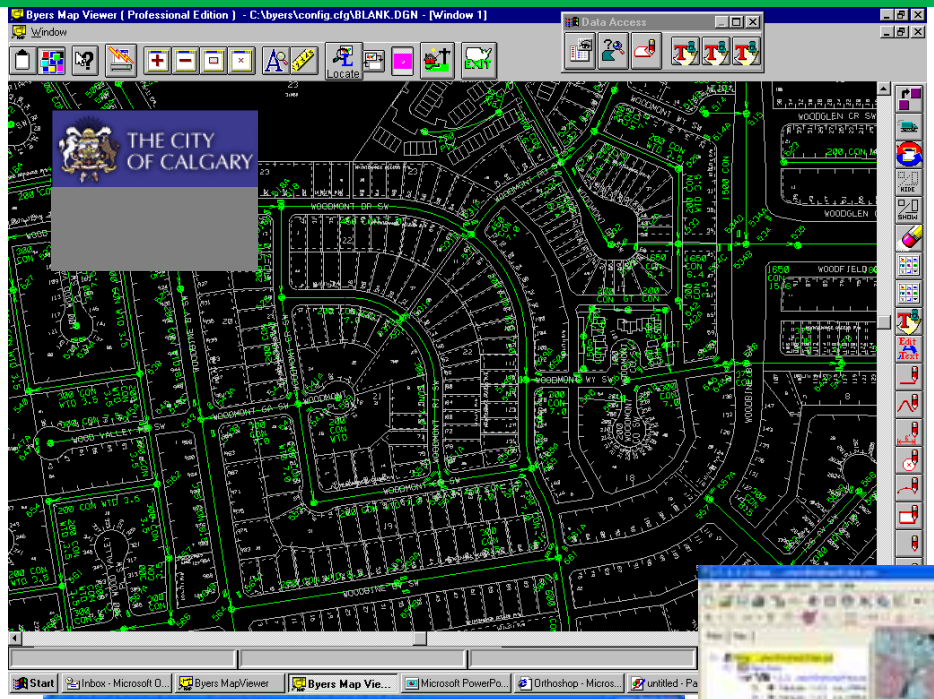
- **Agenda**
  - GIS - systemy informacji przestrzennej
  - Przetwarzanie i analiza danych przestrzennych
    - Modele danych przestrzennych
    - Oracle Spatial
    - Operacje na danych przestrzennych

GIS

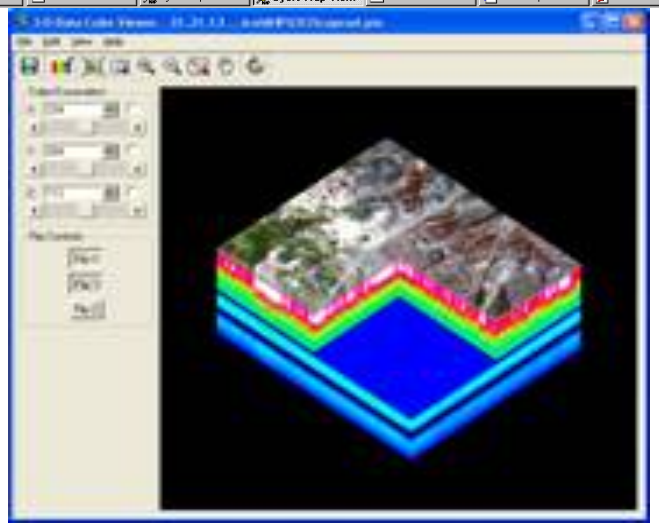
- **GIS – Geographic Information System**
- **Zastosowania**
  - Np..
  - Architektura
  - Planowanie przestrzenne
  - Transport
  - Systemy nawigacji
  - Zarządzanie sytuacjami kryzysowymi

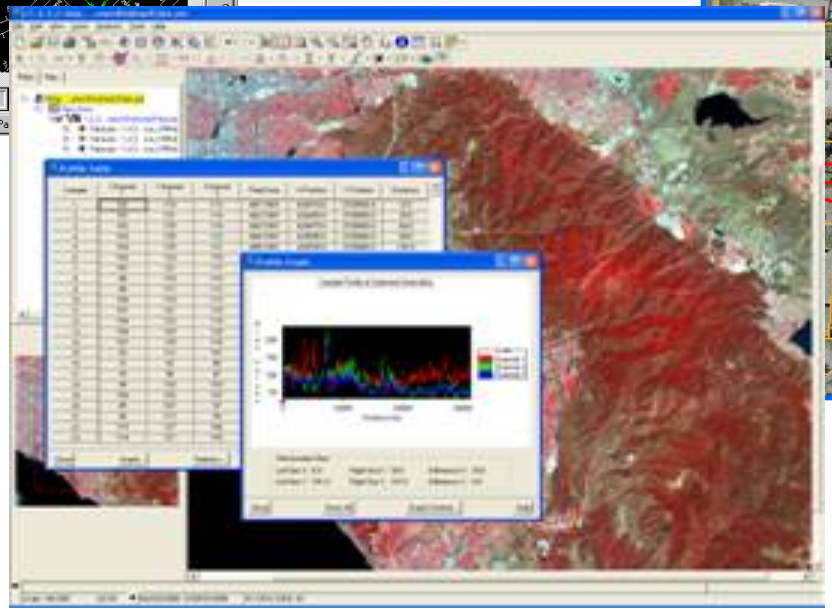
- **Urządzenia mobilne są ważnym elementem GIS**

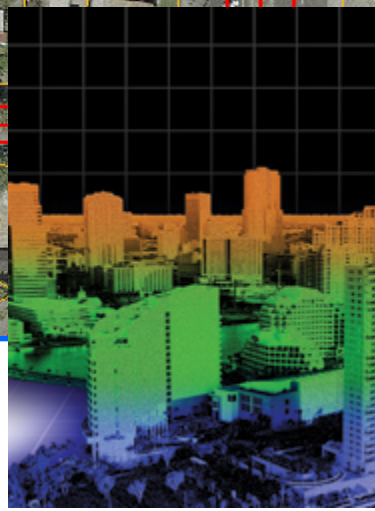
Specjalistczne aplikacje



Geometry
Topology
Georaster
Networks
Geodetic
Long Transactions
3D

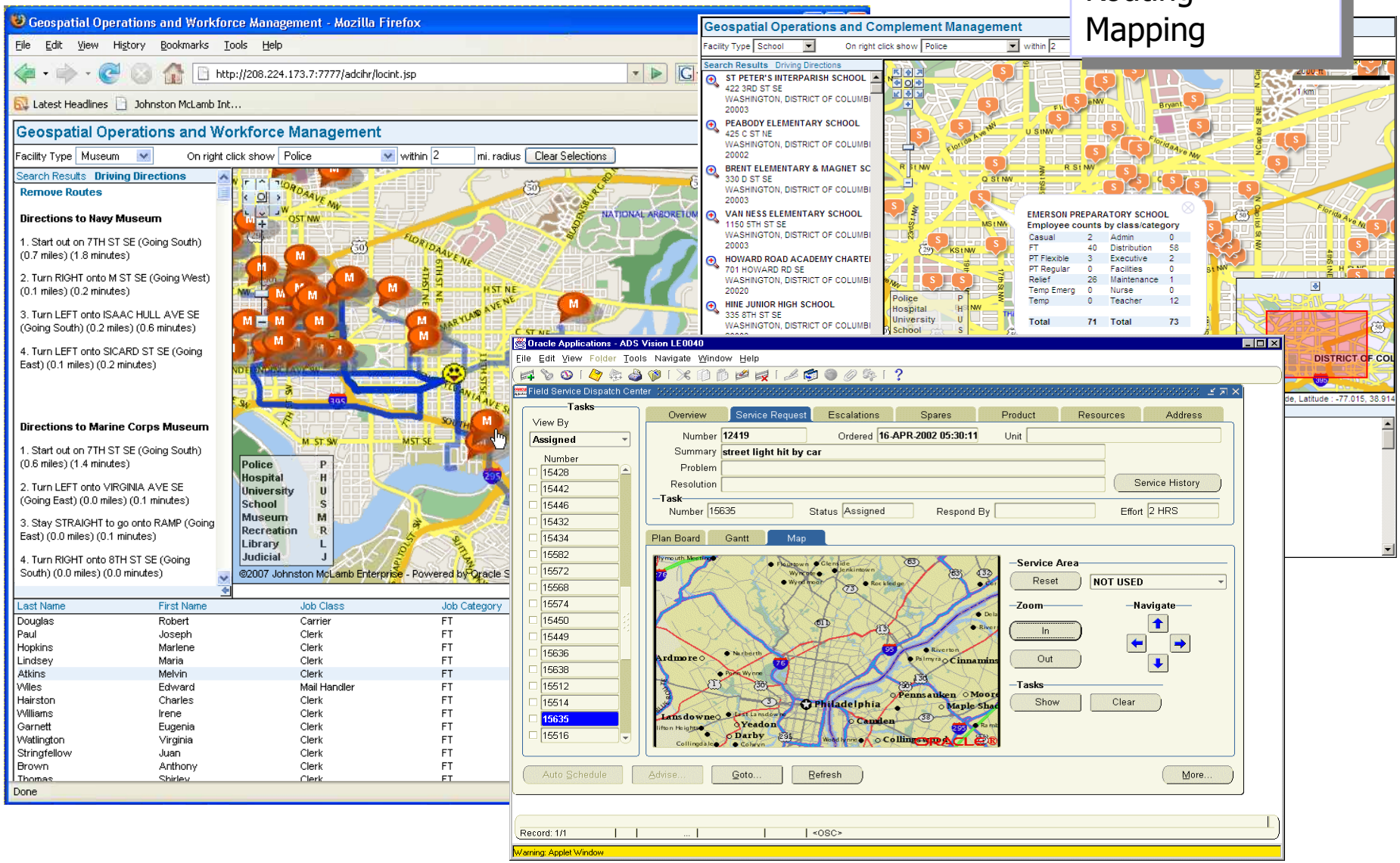






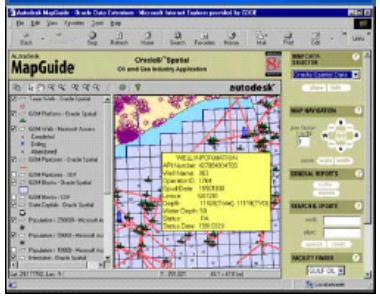
Przetwarzanie danych

Geocoding
Spatial searches
Routing
Mapping



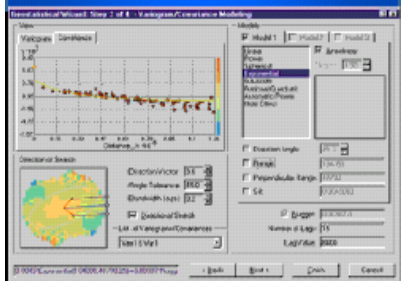
Systemy przetwarzania danych przestrzennych

Autodesk MapGuide

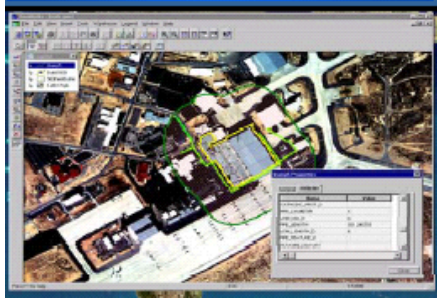


Bentley Microstation

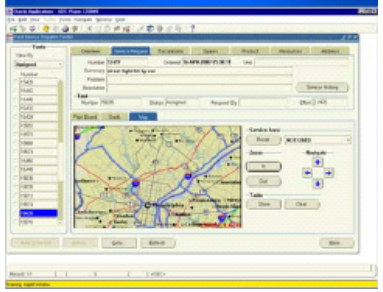
ESRI ArcMap



Intergraph GeoMedia



eSpatial



Oracle Spatial

PCI Geomatics

Skyline
Smallworld
.....

Mapinfo MapXtreme

Bazy danych

- Oracle
- Postgres (PostGis)
- Microsoft SQL Server
- IBM
- IBM
- SQLite, SpatialLite
- MySQL

- MongoDB
- Neo4j

Serwisy mapowe

- OpenStreetMap
- Openlayers
- Google maps
- Bing maps
- Nokia maps
- Esri ArcGis

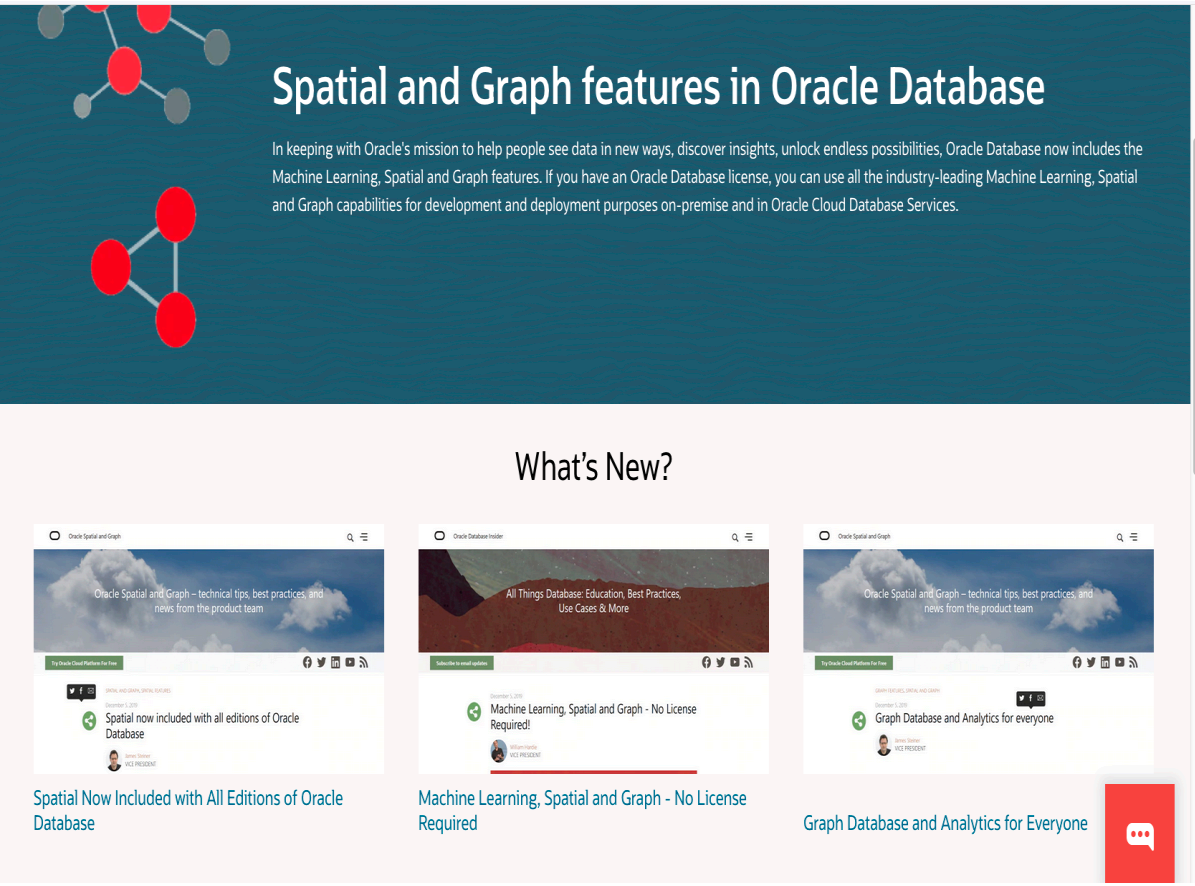
- API, sdk dla różnych systemów i języków programowania

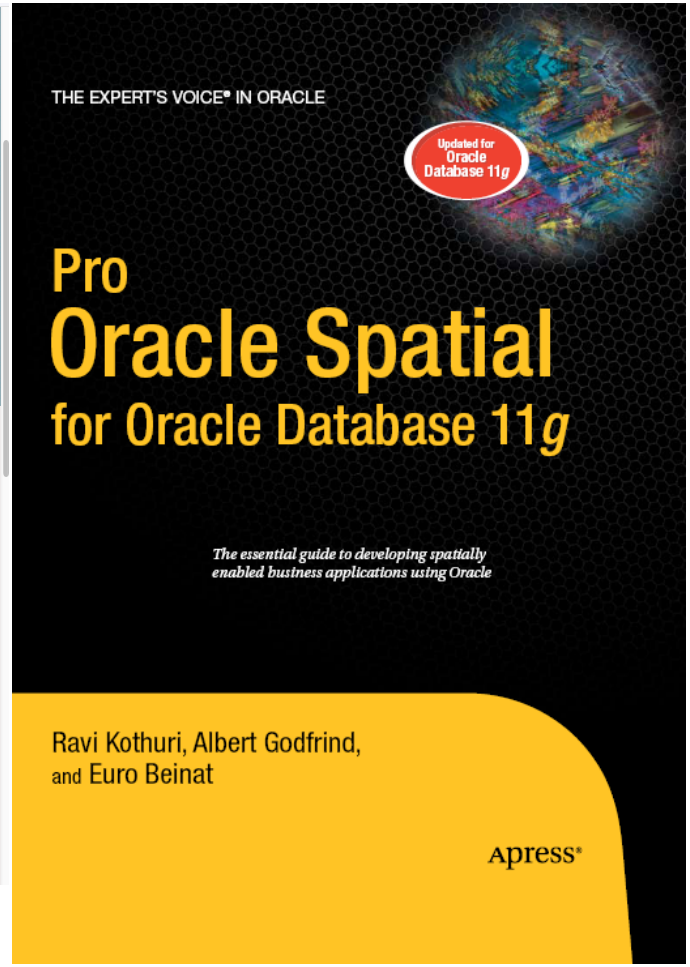
Oprogramowanie

- Oracle Mapviewer
- Esri ArcGIS
- Quantum GIS
- Bentley MicroStation
- I wiele innych

Oracle - zasoby

- **https://www.oracle.com/database/technologies/spatialandgraph.html**



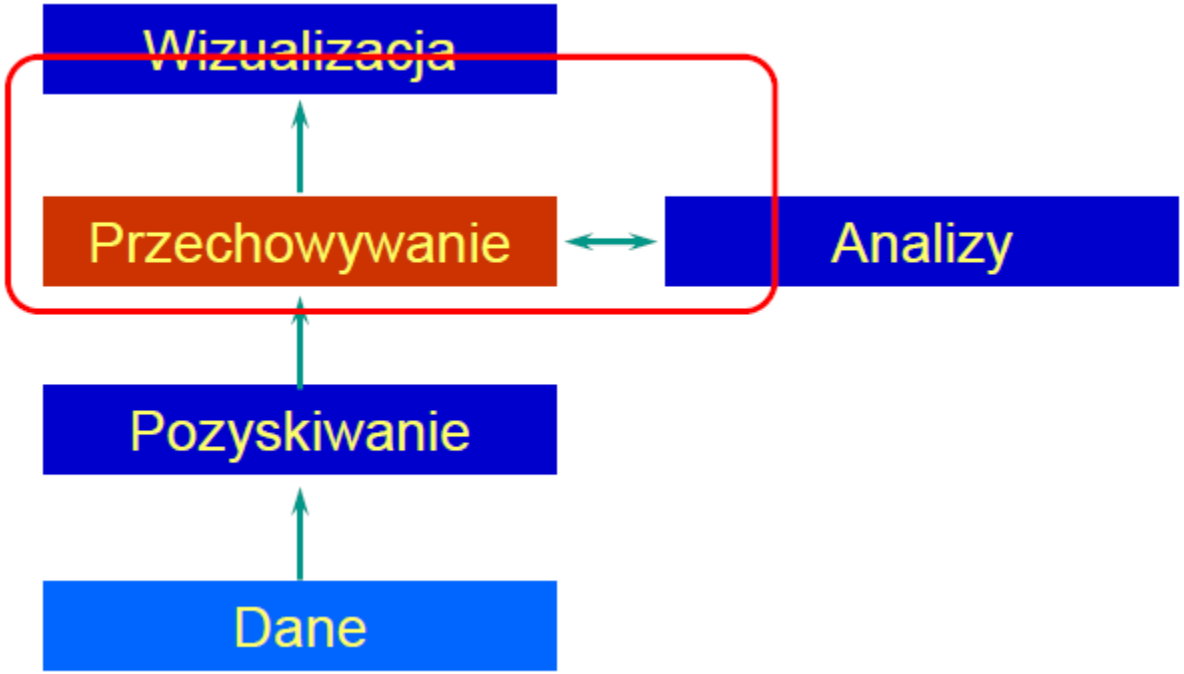


Examples, white papers, downloads, discussion forum, sample data ....

Oracle – maszyny wirtualne

- http://www.oracle.com/technetwork/community/developer-vm/index.html

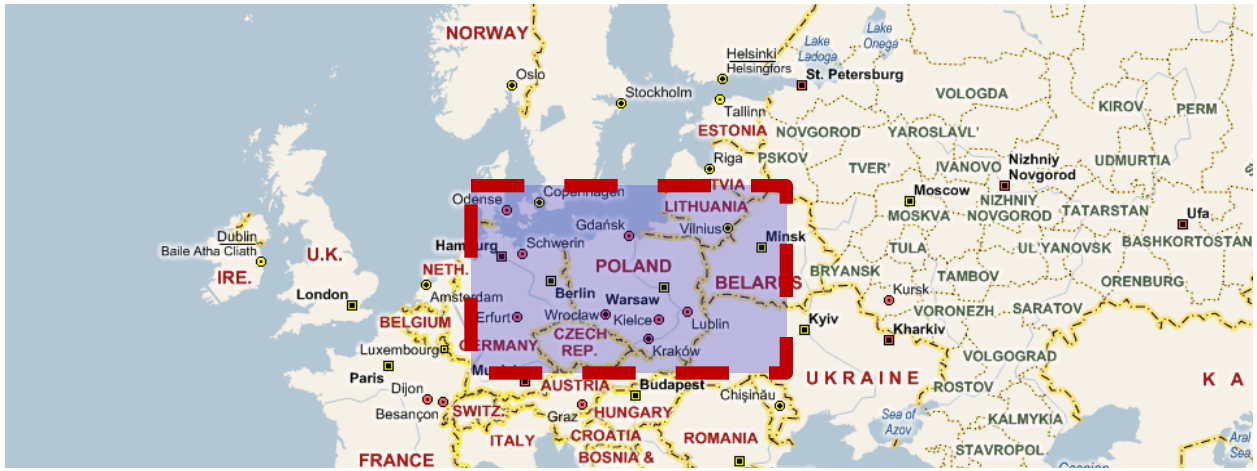
Przetwarzanie danych przestrzennych



Przechowywanie danych

```
CREATE TABLE Customers (
  Name varchar(32),
  Address varchar(255),
  Lat float,
  Long float
);
```

```
SELECT * FROM Customers
WHERE Lat BETWEEN LtMin AND LtMax
AND Long BETWEEN LnMin AND LnMax
```

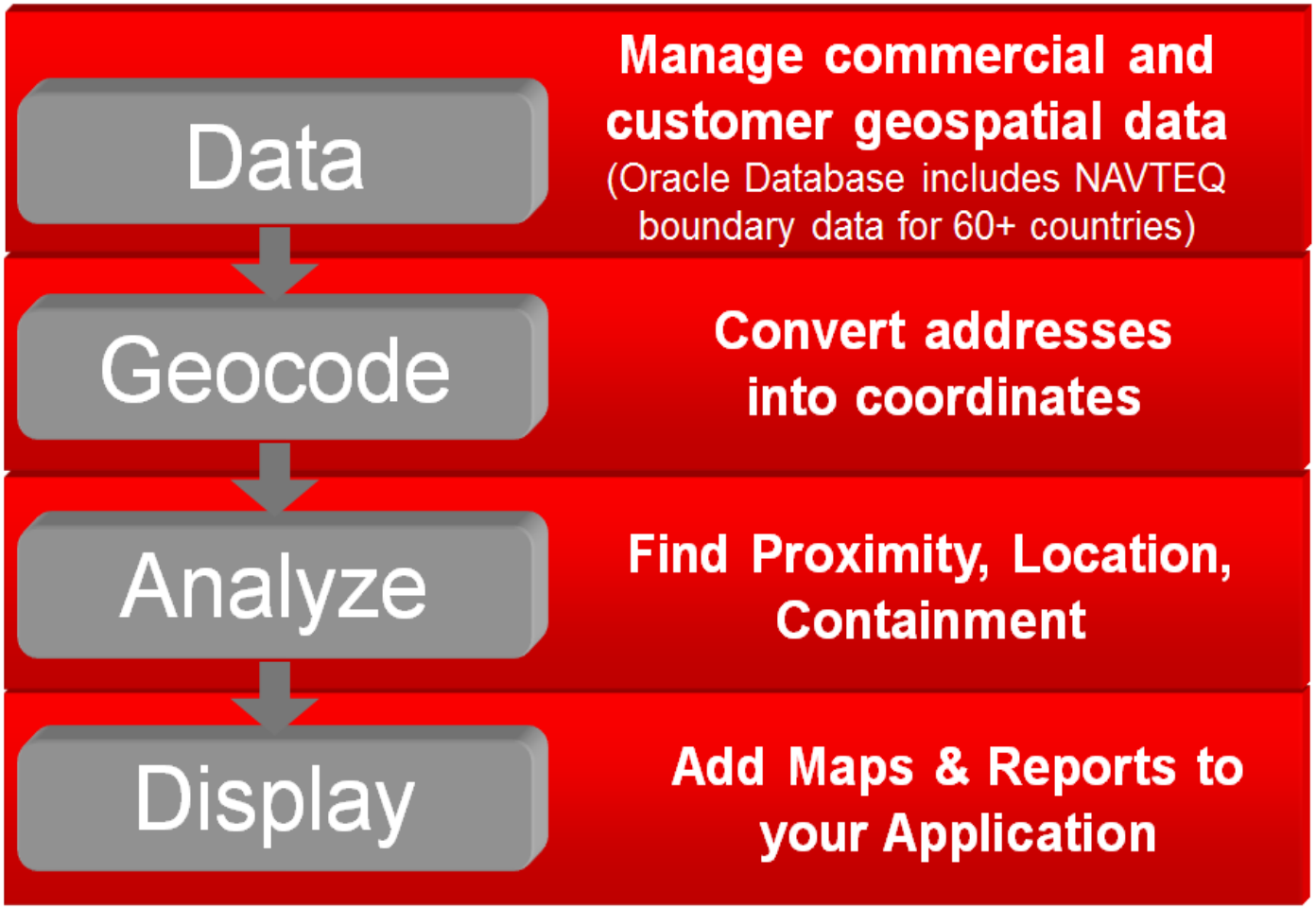


Problemy

- Kształt ziemi
- Odwzorowanie na płaszczyznę
- Układy współrzęnych
- Obliczanie odległości
- Badanie zależności między obiektami
- Badanie relacji przestrzennych

- Rozwiązanie
  - Wprowadzenie typów geometrycznych
  - Operacje na geometriach

Problemy do rozwiązania



Dane przestrzenne

- **Tabele**

Worksheet | Query Builder

```
select * from world_countries

select * from us_states

select * from us_cities
```

Query Result

SQL | Fetched 50 rows in 0,141 seconds

	ID	CITY	STATE_ABRV	POP90	RANK90	LOCATION
1	1	New York	NY	7322564	1	[MDSYS.SDO_GEOMETRY]
2	2	Los Angeles	CA	3485398	2	[MDSYS.SDO_GEOMETRY]
3	3	Chicago	IL	2783726	3	[MDSYS.SDO_GEOMETRY]
4	4	Houston	TX	1630553	4	[MDSYS.SDO_GEOMETRY]
5	5	Philadelphia	PA	1585577	5	[MDSYS.SDO_GEOMETRY]
6	6	San Diego	CA	1110549	6	[MDSYS.SDO_GEOMETRY]
7	7	Detroit	MI	1027974	7	[MDSYS.SDO_GEOMETRY]
8	8	Dallas	TX	1006877	8	[MDSYS.SDO_GEOMETRY]
9	9	Phoenix	AZ	983403	9	[MDSYS.SDO_GEOMETRY]
10	10	San Antonio	TX	935933	10	[MDSYS.SDO_GEOMETRY]
11	11	San Jose	CA	782248	11	[MDSYS.SDO_GEOMETRY]
12	12	Indianapolis	IN	741952	12	[MDSYS.SDO_GEOMETRY]
13	13	Baltimore	MD	736014	13	[MDSYS.SDO_GEOMETRY]
14	14	San Francisco	CA	723959	14	[MDSYS.SDO_GEOMETRY]

Worksheet | Query Builder

```
select * from world_countries

select * from us_states

select * from us_cities
```

Query Result

SQL | Fetched 50 rows in 0,102 seconds

	ID	STATE	STATE_ABRV	FIPSST	TOTPOP	PCTSMPLD	LANDSQMI
1	21	Maryland	MD	24	4781468	14,2988	9772,9609
2	22	Massachusetts	MA	25	6016424	13,3984	7836,6093
3	23	Michigan	MI	26	9295297	18,6992	56799,457
4	24	Minnesota	MN	27	4375096	23,7968	79602,8125
5	25	Mississippi	MS	28	2573216	15,5	46905,625
6	26	Missouri	MO	29	5117073	16,8984	68886,25
7	27	Montana	MT	30	799065	18,7968	145531,375
8	28	Nebraska	NE	31	1578385	20,7968	76864,4375
9	1	Alabama	AL	01	4040586	15,5996	50741,5312
10	2	Alaska	AK	02	550043	20,1992	570275,5
11	3	Arizona	AZ	04	3665228	12,7988	113622,6875
12	4	Arkansas	AR	05	2350724	17,5976	52066,375
13	5	California	CA	06	29760016	12,6992	155946,375

Dane przestrzenne

Worksheet | Query Builder

```
select * from world_countries

select * from us_states

select * from us_cities
```

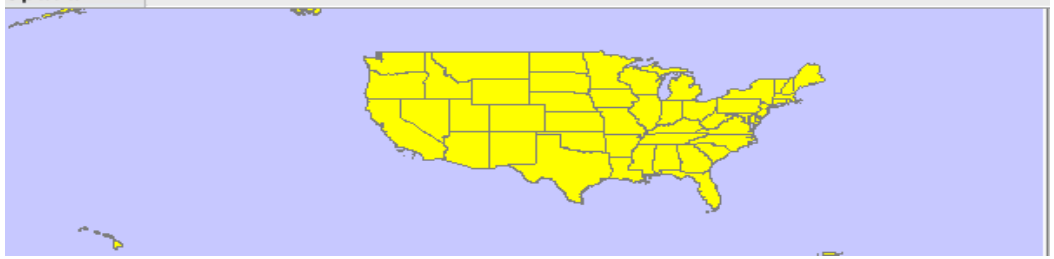
Query Result

SQL | Fetched 50 rows in 0,02 seconds

	ID	STATE	STATE_ABRV	FIPSST	TOTPOP	PC
1	21	Maryland	MD	24	4781468	14
2	22	Massachusetts	MA	25	6016424	13
3	23	Michigan	MI	26	9295297	18
4	24	Minnesota	MN	27	4375096	23
5	25	Mississippi	MS	28	2573216	
6	26	Missouri	MO	29	5117073	16
7	27	Montana	MT	30	799065	18
8	28	Nebraska	NE	31	1578385	20
9	1	Alabama	AL	01	4040586	15
10	2	Alaska	AK	02	550043	20
11	3	Arizona	AZ	04	3665228	12
12	4	Arkansas	AR	05	2350724	17
13	5	California	CA	06	29760016	12

Map View

Spatial 1



Warstwa przestrzenna

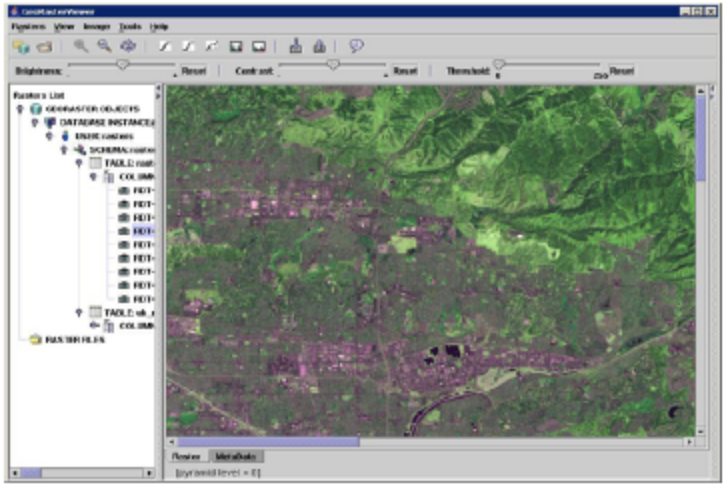
- **Warstwa przestrzenna**
  - Jest zbiorem geometrii posiadających podobny zestaw atrybutów (np. związanych znaczeniowo)
- **Przykłady warstw przestrzennych:**
  - Mapy krajów, miast
  - Sieci dróg,
  - Sieci wodociągów,
  - Obszary leśne, obszary zurbanizowane,
  - Sieci sklepów
- **W bazie danych warstwa jest z reguły tożsama z pojedynczą kolumną w tabeli.**

Dane przestrzenne

- **Model rastrowy**
- **Model wektorowy**

Model rastrowy

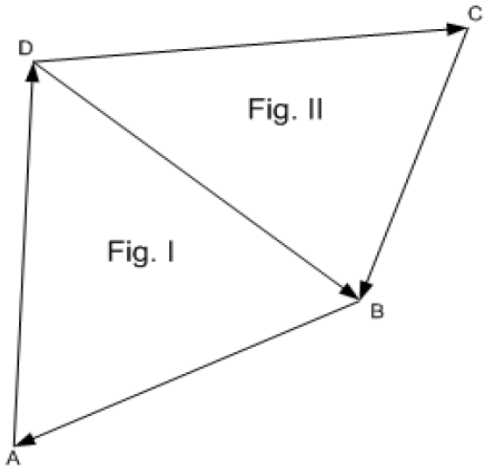
- **Rastrowy model przechowywania danych przestrzennych opiera się na zapisie macierzowym**
  - informacje w nim zapisane reprezentowane są przez wartości numeryczne umieszczone w komórkach macierzy.
  - liczby umieszczone w macierzy opisującej przestrzeń mogą reprezentować dowolne informacje
    - Np.. Wysokość nad poziomem morza, wilgotność, gęstość zaludnienia, stopę bezrobocia , dowolna wielkość możliwa do przedstawienia przy pomocy liczb.

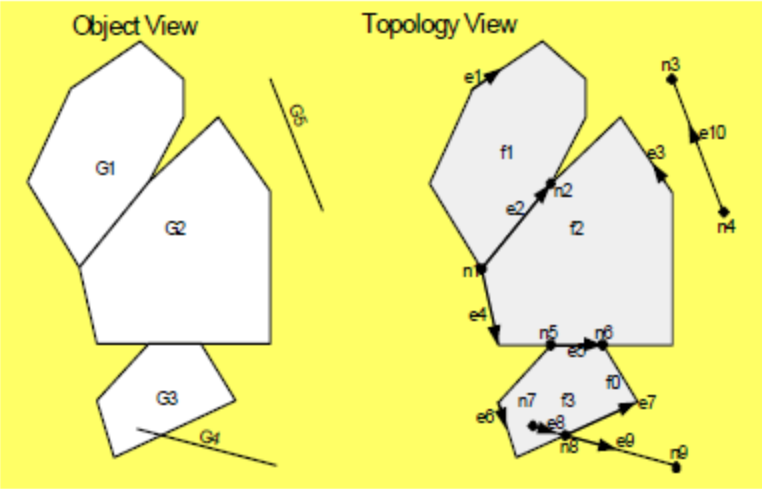


- **Jedna komórka macierzy określana jest mianem piksela**

Model wektorowy

- **Figury geometryczne opisane są przy pomocy punktów w obranym układzie odniesienia**
- **Punkty są ze sobą powiązane tworząc obiekty geometryczne (opisujące kształty)**





Przetwarzanie danych przestrzennych

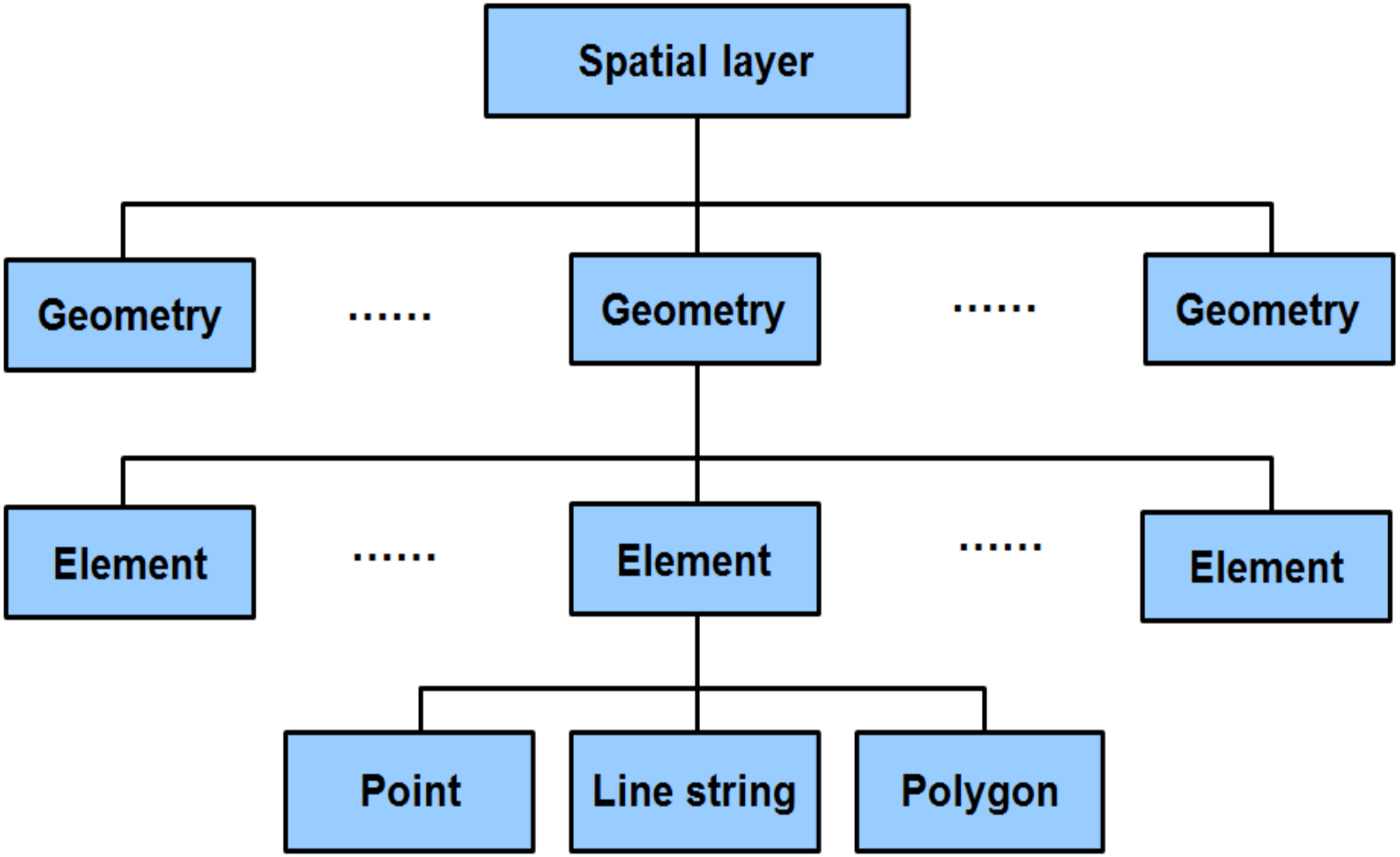
- **Rozwiązania na przykładzie Oracle Spatial**

- **Reprezentacja danych przestrzennych**
- **Analiza danych przestrzennych**

Warstwy

- **Pojęcie warstwy przestrzennej**
- **Tabela**
  - Atrybuty
  - Kolumna typu przestrzennego

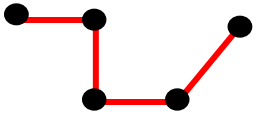
Model wektorowy - hierarcha obiektów



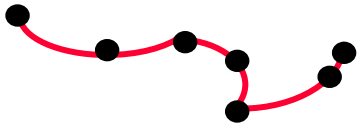
Geometrie

Point

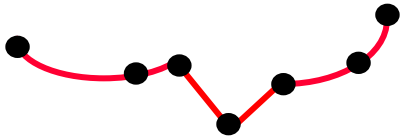
Line string



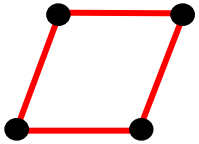
Arc line string



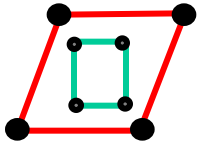
Compound line string



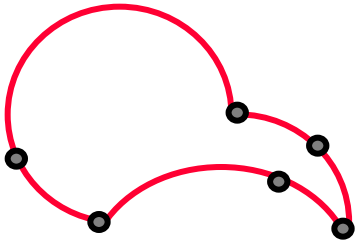
Polygon



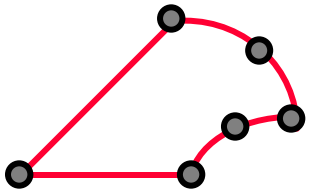
Polygon with one or more holes



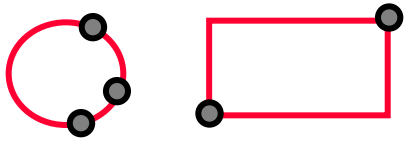
Arc polygon



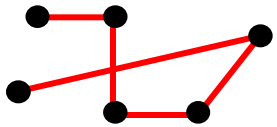
Compound polygon



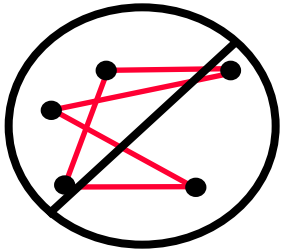
Optimized polygons



Self-crossing line strings valid



Self-crossing polygons not valid



Geometrie

Geometry 1
California

Geometry 2
Texas

Geometry 3
Florida

Geometry 4
Hawaii

Typ obiektowy SDO_GEOMETRY

- Typ SDO_GEOMETRY – pozwalający na zdefiniowanie i przechowywanie geometrii
- Przechowywany w schemacie użytkownika MDSYS

```
MDSYS.SDO_GEOMETRY AS OBJECT (
SDO_GTYPE NUMBER,
SDO_SRID NUMBER,
SDO_POINT SDO_POINT_TYPE,
SDO_ELEM_INFO MDSYS.SDO_ELEM_INFO_ARRAY,
SDO_ORDINATES MDSYS.SDO_ORDINATE_ARRAY);
```

```
CREATE TABLE "COUNTRY_BOUNDARIES"
(
"FIPS_CNTRY" VARCHAR2(2),
"CNTRY_NAME" VARCHAR2(40),
"GEOM" "MDSYS"."SDO_GEOMETRY”
);
```

SDO_GEOMETRY - atrybut SDO_GTYPE

- **SDO_GTYPE – określa typ geometrii w formacie dltt**
  - d – liczba wymiarów wykorzystywanych do opisu geometrii
  - l – 0 –
  - tt – typ geometrii.
- **Dopuszczalne wartości to:**
  - dl00 – UNKNOWN_GEOMETRY – geometria ignorowana przez funkcje i operatory przestrzenne
  - dl01 – POINT – geometria składaj ca się z jednego punktu.
  - dl02 – LINESTRING – geometria składaj ca się z jednego ciągu linii.
  - dl03 – POLYGON – jeden wielokąt.
  - dl04 – COLLECTION – różnorodna kolekcja elementów.
  - dl05 – MULTIPOINT – geometria składaj ca się z wielu punktów.
  - dl06 – MULTILINESTRING – wiele ciągów linii.
  - dl07 – MULTIPOLYGON – wiele rozłącznych wielokątów lub powierzchni.

```
SDO_GEOMETRY AS
SDO_GTYPE
SDO_SRID
SDO_POINT
SDO_ELEM_INFO
SDO_ORDINATES
```

```
SDO_GEOMETRY(2007, 8307, NULL,
SDO_ELEM_INFO_ARRAY(1, 1003, 1, 277, 1003, 1, 421, 1003, 1, 609,
1003, 1, 695, 1003, 1),
SDO_ORDINATE_ARRAY(12.603676, 50.408755, 12.620972, 50.411942,
12.671389, …) )
```

SDO_GEOMETRY - atrybut SDO_SRID

- **SDO_SRID – odniesienia (odwzorowanie)**
  - Wartość pusta oznacza, że żaden układ odniesienia nie jest wykorzystywany, współrzędne są współrzędnymi kartezjańskimi.
- **W bazie danych Oracle dostępnych jest ok 4,5 tys. układów odniesienia**

```
SDO_GEOMETRY AS
SDO_GTYPE
SDO_SRID
SDO_POINT
SDO_ELEM_INFO
SDO_ORDINATES
```

8307 Longitude / Latitude (WGS 84)

```
SDO_GEOMETRY(2007, 8307, NULL,
SDO_ELEM_INFO_ARRAY(1, 1003, 1, 277, 1003, 1, 421, 1003, 1, 609,
1003, 1, 695, 1003, 1),
SDO_ORDINATE_ARRAY(12.603676, 50.408755, 12.620972, 50.411942,
12.671389, …) )
```

SDO_GEOMETRY - atrybut SDO_ELEM_INFO

- **SDO_ELEM_INFO zawiera opis poszczególnych elementów wchodzących w skład geometrii.**
- **Każdy element opisywany jest przez trzy parametry:**
  - SDO_STARTING_OFFSET - pozycja pierwszej koordynaty w SDO_ORDINATES
  - SDO_ETYPE – definiujący typ elementu
  - SDO_INTERPRETATION – interpretacja, posiada różne znaczenie w zależno ci od typu elementu

```
SDO_GEOMETRY AS
SDO_GTYPE
SDO_SRID
SDO_POINT
SDO_ELEM_INFO
SDO_ORDINATES
```

```
SDO_GEOMETRY(2007, 8307, NULL,
SDO_ELEM_INFO_ARRAY(1, 1003, 1, 277, 1003, 1, 421, 1003, 1,
609, 1003, 1, 695, 1003, 1),
SDO_ORDINATE_ARRAY(12.603676, 50.408755, 12.620972, 50.411942,
12.671389, …) )
```

SDO_ETYPE i SDO_INTERPRETATION

Number	Element Type	Interpretation
0	UNKNOWN_ELEMENT	
1	POINT	# of points in collection; or 0 - oriented point
2	LINESTRING	1 - Straight lines 2 - Circular arcs
3 1003 2003	POLYGON (Outer) (Inner)	1 - Straight lines 2 - Circular arcs 3 - Optimized rectangle 4 - Circle
4	COMPOUND LINESTRING	# of type 2 subelements that make up the line string
5 1005 2005	COMPOUND POLYGON (Outer) (Inner)	# of type 2 subelements that make up the polygon

Warstwa przestrzenna - podsumowanie

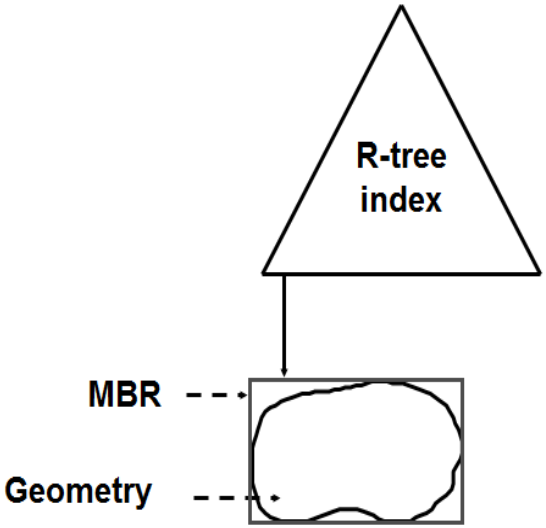
```
--metadane
INSERT INTO USER_SDO_GEOM_METADATA
VALUES (
'geometrie1', 'geometria',
MDSYS.SDO_DIM_ARRAY(
MDSYS.SDO_DIM_ELEMENT('X', 0, 20, 0.005),
MDSYS.SDO_DIM_ELEMENT('Y', 0, 20, 0.005) ),
NULL );


--indeks
CREATE INDEX figura_spatial_idx
ON geometrie1(geometria)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Indeksowanie

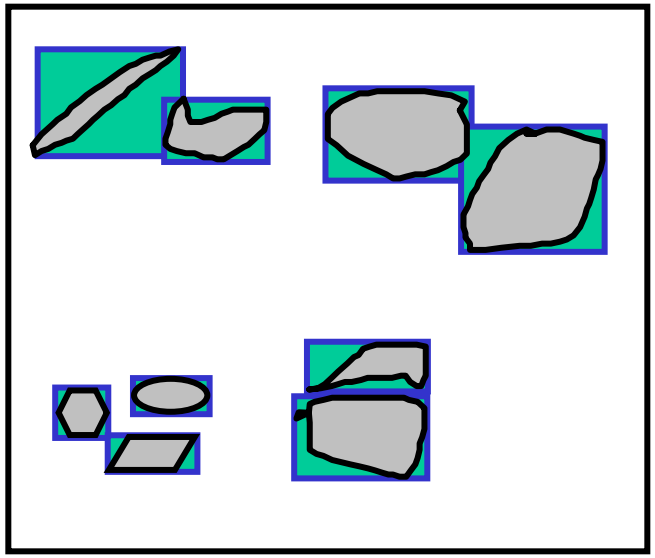
- **Indeks przestrzenny**
- **Umożliwia korzystanie z operatorów przestrzennych**
  - Użycie niektórych operatorów nie jest możliwe bez indeksu przestrzennego

```
create index COUNTRY_BOUNDARIES_IDX
on COUNTRY_BOUNDARIES(GEOM)
INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

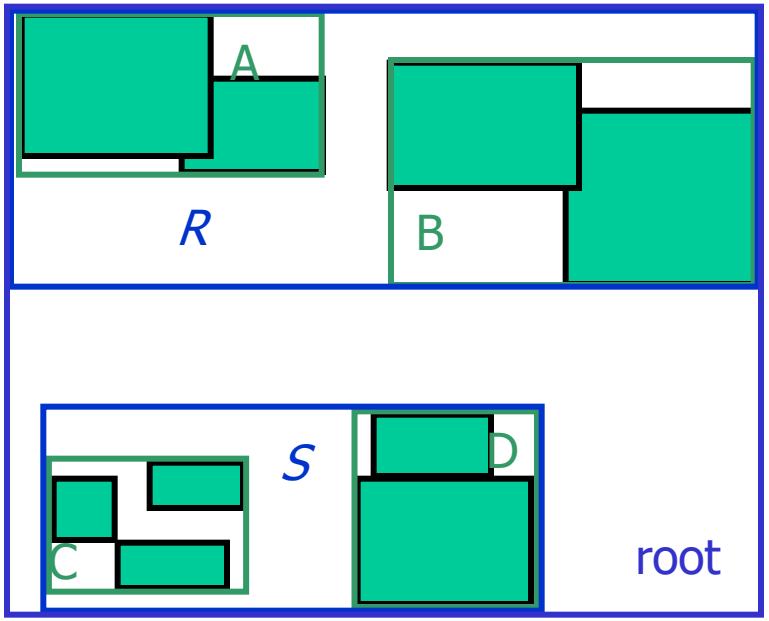
Indeksowanie – MBR, R-tree

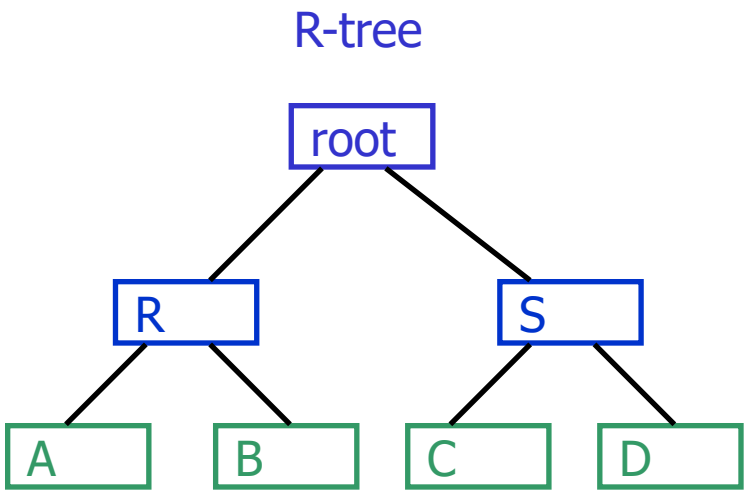


Leaf nodes of R-tree store
<MBR, geometry pointer>



Indeksowanie – MBR, R-tree

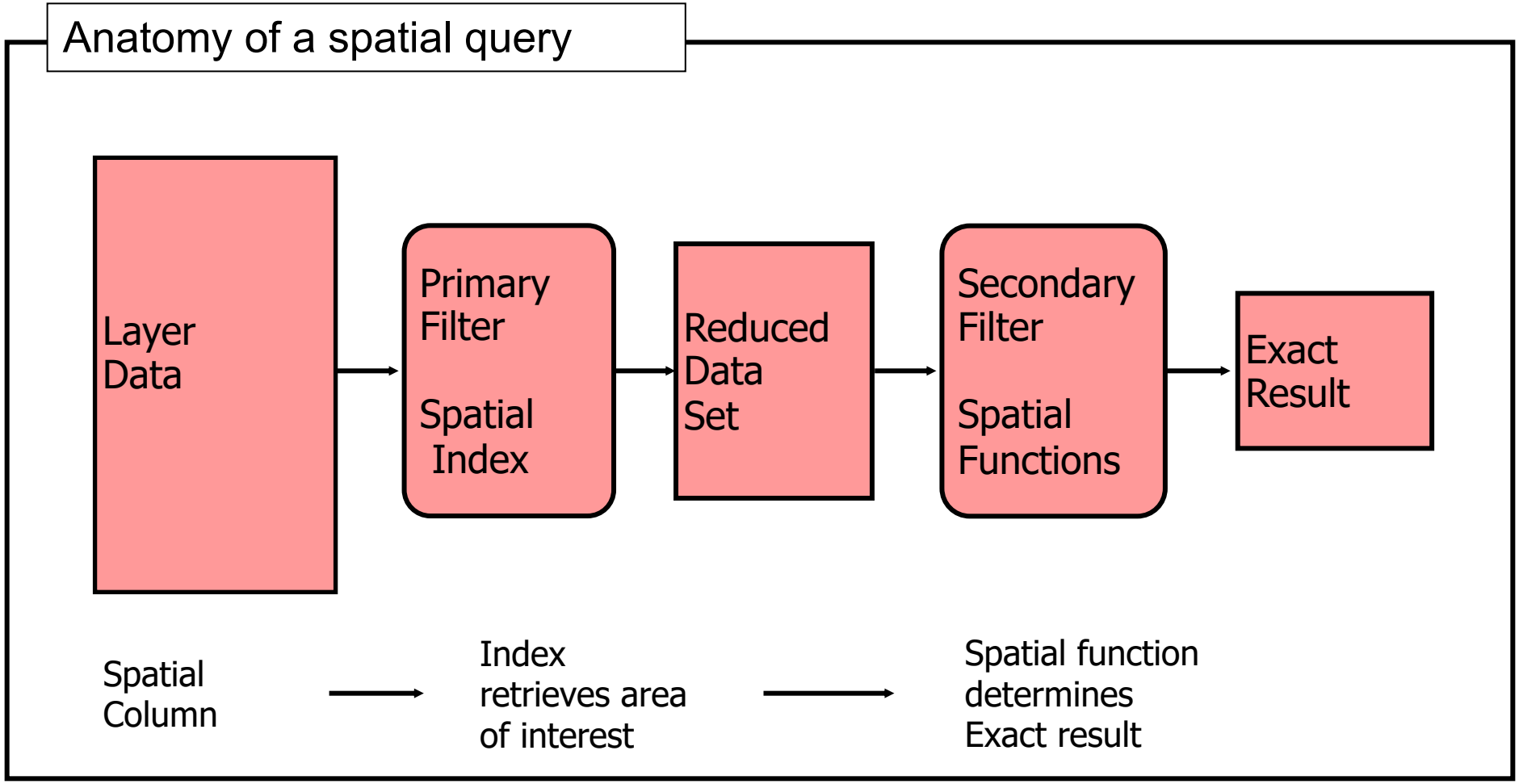




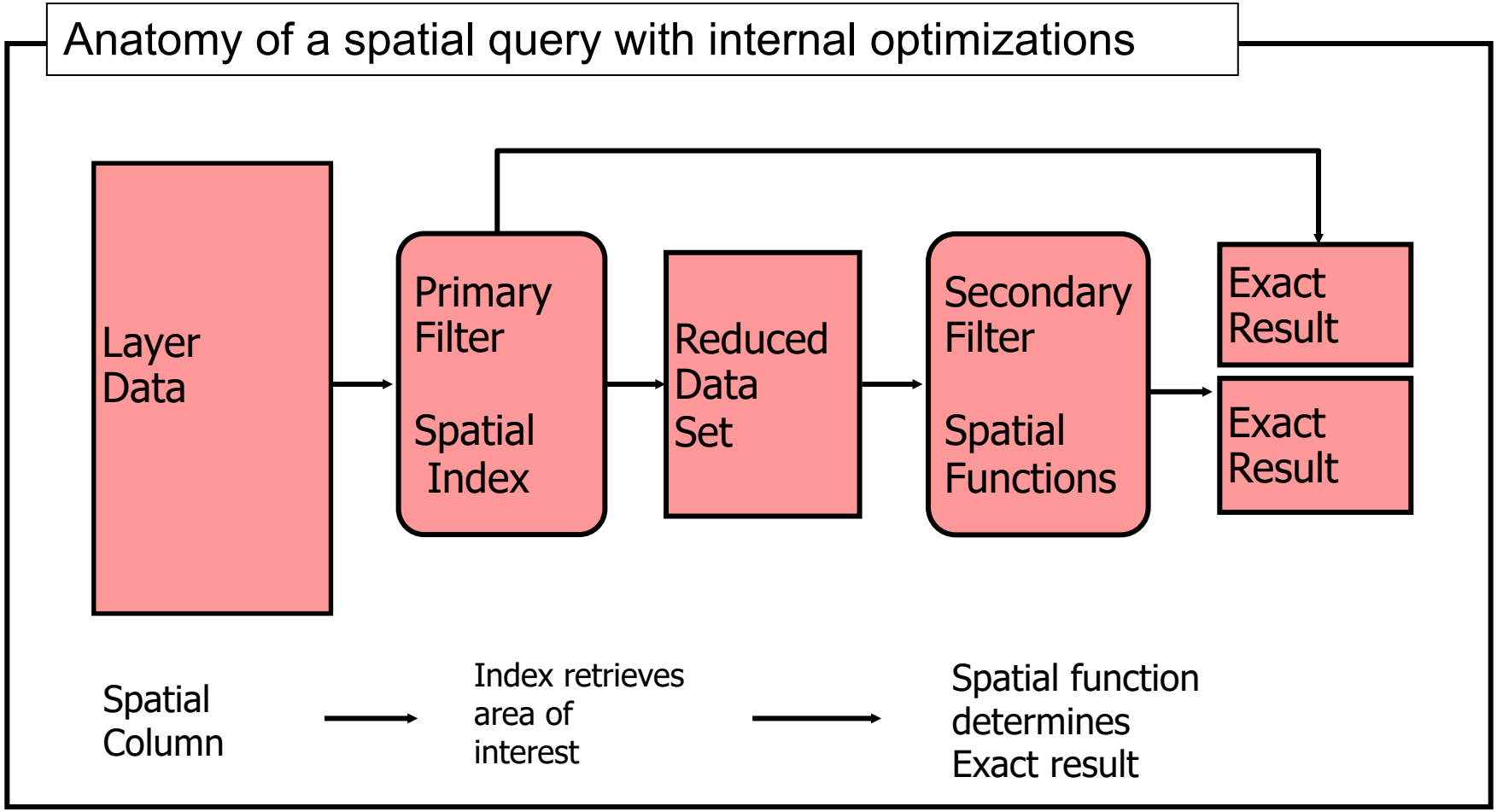
Zapytania przestrzenne

- **Zapytania przestrzenne przetwarzane są w 2 fazach.**
  - Podczas pierwszej fazy z pełnego zbioru danych na podstawie aproksymacji opartej na indeksie przestrzennym wybierane są te obiekty, które potencjalnie mogą spełnia warunki zapytania s to tzw. kandydaci.
  - W fazie drugiej ze zbioru kandydatów wyznaczane są za pomocą metod dokładnych te obiekty, które rzeczywiście spełniaj zadany warunek

Przetwarzanie danych

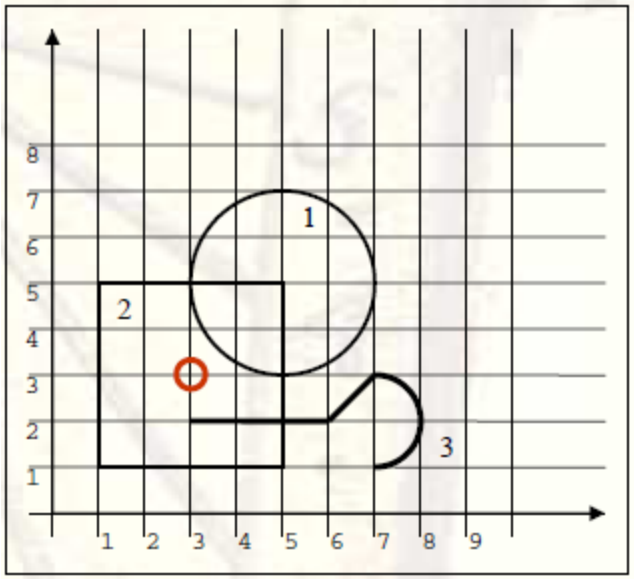


Przetwarzanie danych



Przykład

- Operator SDO_FILTER, który wykorzystuje jedynie pierwszą faz zapytania, czyli daje w wyniku zbiór "kandydatów",
- Operator SDO_RELATE, który wykorzystuje obie fazy zapytania, czyli dokonuje dodatkowej weryfikacji zbioru kandydatów, będzie zwróci inny wynik.




```
select ID
from FIGURY
where SDO_FILTER(KSZTALT,
MDSYS.SDO_GEOMETRY(2001,null,
MDSYS.SDO_POINT_TYPE(3,3,null),
null,null),'querytype=WINDOW') = 'TRUE';
ID
----------
3
2
1
```

```
select ID
from FIGURY
where SDO_RELATE(KSZTALT,
MDSYS.SDO_GEOMETRY(2001,null,
MDSYS.SDO_POINT_TYPE(3,3,null),
null,null),
'mask=ANYINTERACT
querytype=WINDOW') = 'TRUE';
ID
----------
2
```

Przetwarzanie danych przestrzennych

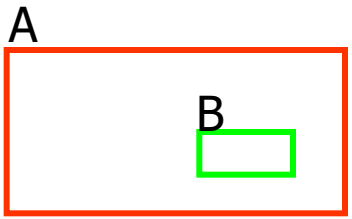
- **Dane przestrzenne można przetwarzać za pomoc bardzo bogatego zbioru operatorów i funkcji.**
- **Wyróżniamy:**
  - Operatory przestrzenne (Spatial Operators)
  - Funkcje geometryczne (Geometry Functions)
  - Przestrzenne funkcje grupujące (Spatial Aggregate Functions)

Relacje przestrzenne

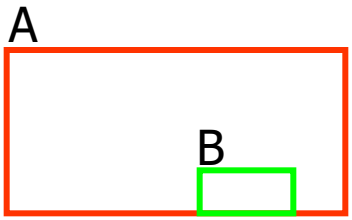
- **ANYINTERACT**
  obiekty nie są rozłączne
- **CONTAINS**
  drugi obiekt całkowicie zawiera się wewnątrz pierwszego obiektu, granice obiektów się nie stykają
- **COVEREDBY**
  obiekt jest całkowicie zawarty wewnątrz drugiego obiektu, granice obiektów stykaj się w jednym lub wielu punktach.
- **COVERS**
  drugi obiekt jest całkowicie wewnątrz pierwszego obiektu, granice nakładaj się w jednym lub wielu miejscach.
- **DISJOINT**
  obiekty nie maj wspólnych granic ani wewnętrznych elementów.
- **EQUAL**
  obiekty współdziel każdy punkt swoich granic i wnętrza włączaj c w to dziury wewnątrz obiektów.
- **INSIDE**
  obiekt całkowicie zawiera się wewnątrz drugiego obiektu, granice nie stykaj się.
- **OVERLAPBDYDISJOINT**
  obiekty zachodzą na siebie lecz ich granice nie mają części wspólnych.
- **OVERLAPBDYINTERSECT**
  obiekty zachodzą na siebie a ich granice mają części wspólne w jednym lub wielu fragmentach.
- **TOUCH**
  jeśli dwa obiekty współdziel wspólne punkty graniczne, i jednocześnie nie współdzielą żadnych punktów wewnętrznych.

Relacje przestrzenne

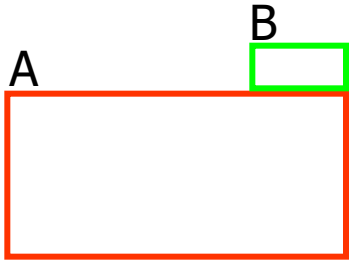
AnyInteract



A Contains B
B Inside A

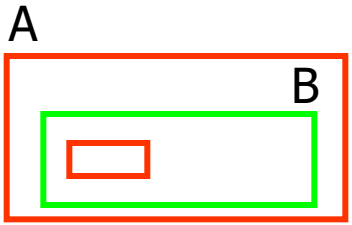


A Covers B
B Coveredby A

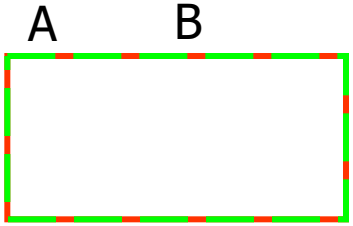


Touch

OverlapBdyIntersect

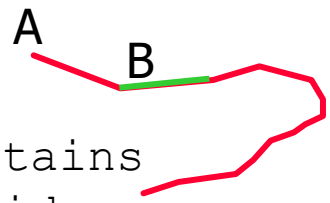


OverlapBdyDisjoint

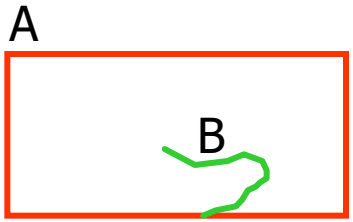


Equal

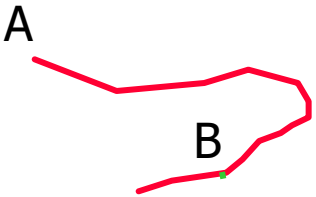
Relacje przestrzenne



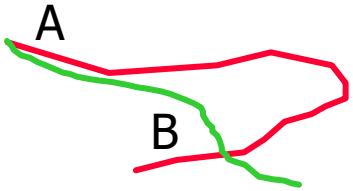
Contains
Inside



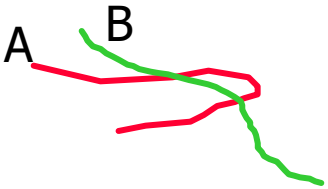
Covers
Coveredby



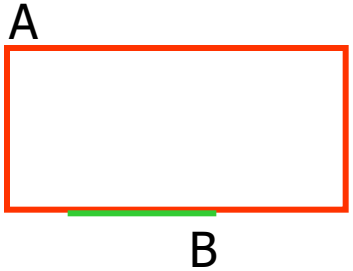
Contains
Inside

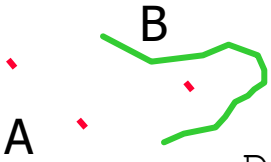


OverlapBdyIntersect



OverlapBdyDisjoint





Disjoint

Przykłady operatorów przestrzennych

- **SDO_FILTER**
  - sprawdza czy dwa obiekty występuj w określonej zależności. Wykorzystuje tylko filtr podstawowy, wymaga indeksu.
- **SDO_NN**
  - identyfikacja najbliższego sąsiedztwa, wymaga indeksu.
- **SDO_NN_DISTANCE**
  - odległości od obiektów zwróconych przez operator SDO_NN.
- **SDO_RELATE**
  - bada występowanie określonej relacji przestrzennej.
- **SDO_WITHIN_DISTANCE**
  - wyznaczanie zbioru geometrii znajdujących się w określonej odległości od obiektu.

- **W przypadku każdego z operatorów, pierwszym argumentem musi być kolumna na której został założony indeks przestrzenny.**

Funkcje geometryczne

- **Wyznaczające relacje pomiędzy dwoma obiektami:**
  - RELATE – określa zależność pomiędzy obiektami,
  - WITHIN_DISTANCE – sprawdza czy obiekty znajduj się w określonej odległości
- **Operacje na dwóch obiektach:**
  - SDO_DISTANCE – oblicza odległo od geometrii
  - SDO_DIFFERENCE – różnica topologiczna dwóch geometrii,
  - SDO_INTERSECTION – część wspólna dwóch geometrii,
  - SDO_UNION – suma topologiczna dwóch geometrii,
  - SDO_XOR – symetryczna różnica dwóch geometrii

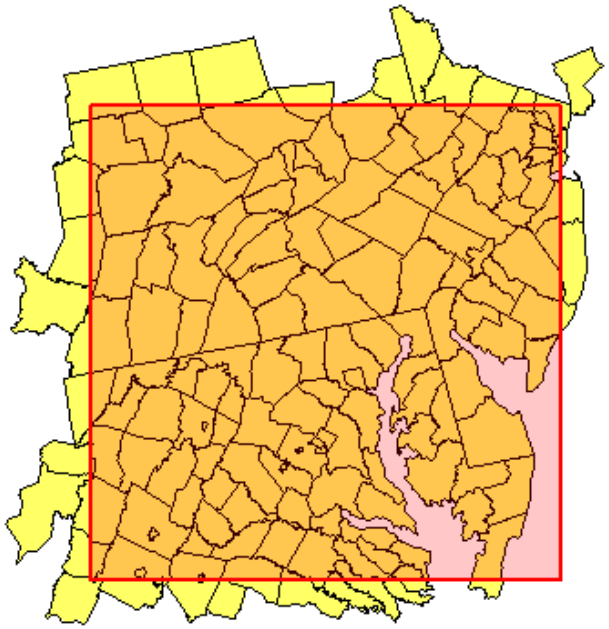
Funkcje geometryczne

- **Operacje na pojedynczych obiektach:**
  - SDO_ARC_DENSIFY – zamienia łuki i okręgi na przybliżone odpowiedniki wyrażone za pomoc linii prostych,
  - SDO_AREA – wyznacza powierzchni dwuwymiarowej geometrii,
  - SDO_BUFFER – generuje wielokąt otaczający geometrię,
  - SDO_CENTROID – zwraca punkt georeferencyjny (tzw.controid) geometrii,
  - SDO_CONVEXHULL – zwraca obiekt reprezentujący powłokę wypukłą (convex hull) geometrii,
  - SDO_LENGTH – wylicza długość obwodu,
  - SDO_MBR – wyznacza najmniejszy prostokąt obejmujący geometrię,

Przykłady

SDO_FILTER - przyklad

- Find all the counties likely to interact with a selected rectangular area

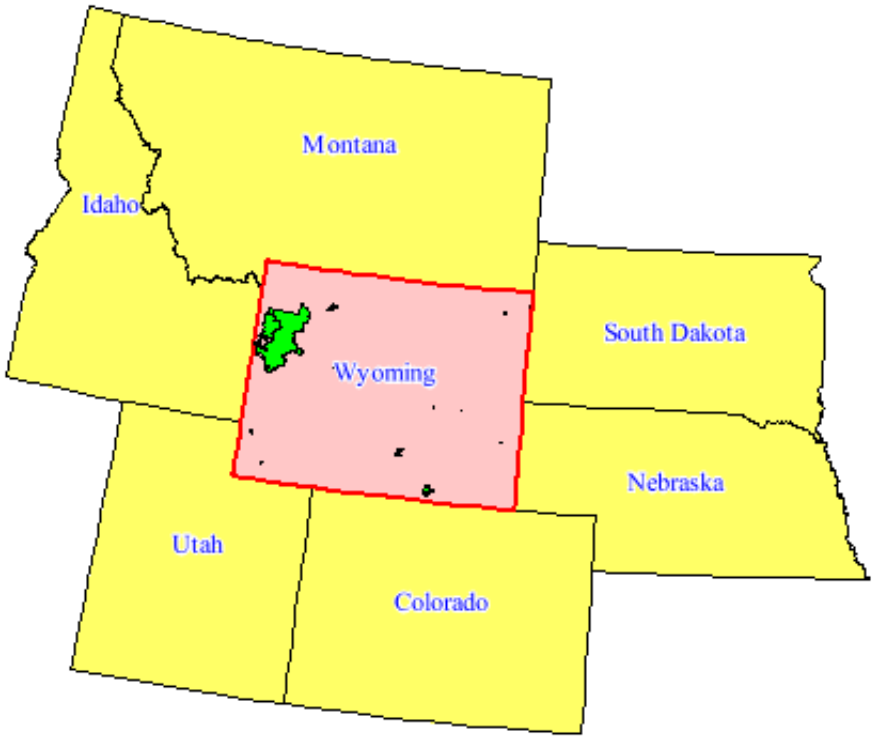


```
SELECT county, totpop, geom
FROM us_counties_p
WHERE sdo_filter (
        geom,
        sdo_geometry (2003, 32775, null,
          sdo_elem_info_array (1,1003,3),
          sdo_ordinate_array (
            1420300,1805461, 1820000,2210000))
        ) = 'TRUE';
```

Przykład

- Find all the parks fully inside the state of Wyoming

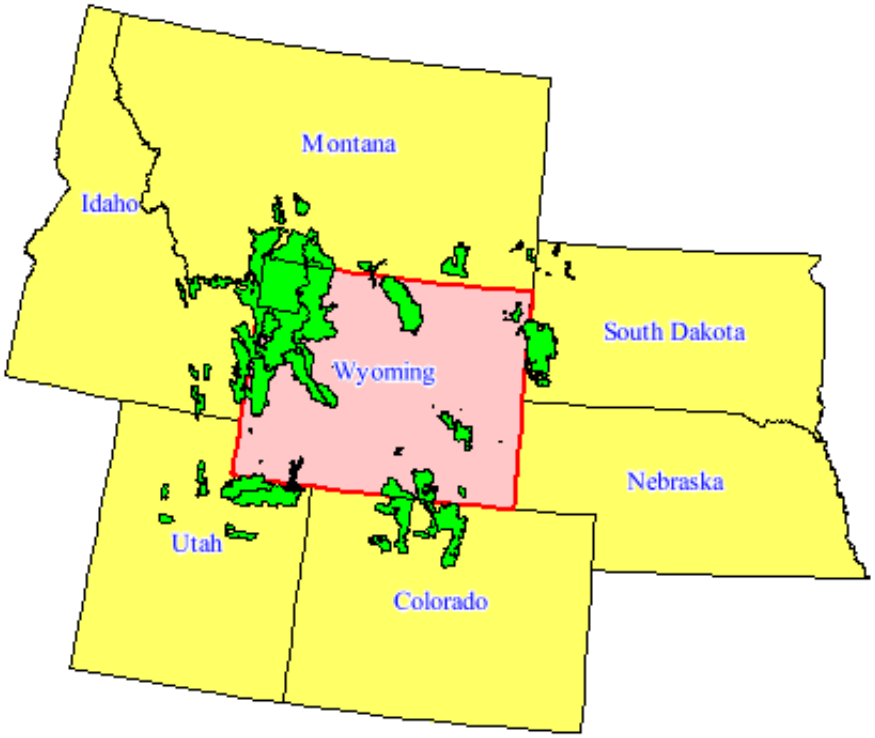
```
SELECT p.id, p.name
  FROM us_parks p, us_states s
 WHERE s.state = 'Wyoming'
   AND SDO_INSIDE (
         p.geom, s.geom
       ) = 'TRUE';
```



Przykład

- Find all the parks that interact with the state of Wyoming

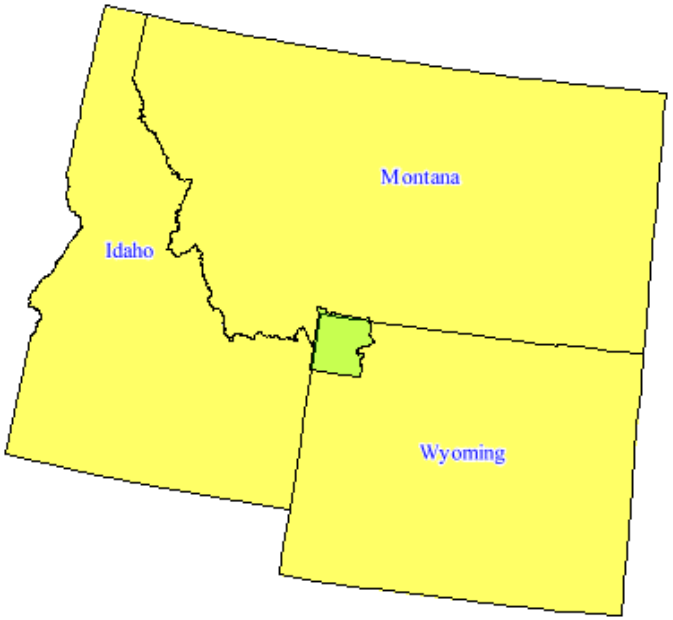
```
SELECT p.id, p.name
  FROM us_parks p, us_states s
 WHERE s.state = 'Wyoming'
   AND SDO_ANYINTERACT (
         p.geom, s.geom
       ) = 'TRUE';
```



Przykład

- Find all the states that interact with the Yellowstone park

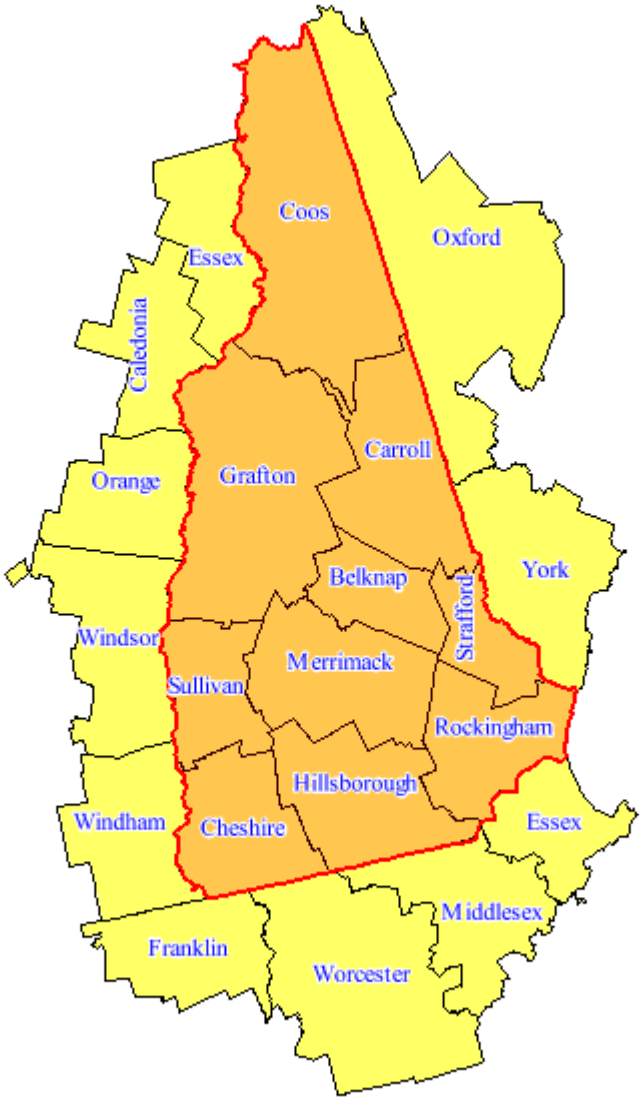
```
SELECT s.state
  FROM us_states s, us_parks p
 WHERE p.name = 'Yellowstone NP'
   AND SDO_ANYINTERACT (
         s.geom, p.geom
       ) = 'TRUE';
```



Przykład

- **Find all the counties in the state of New Hampshire:**

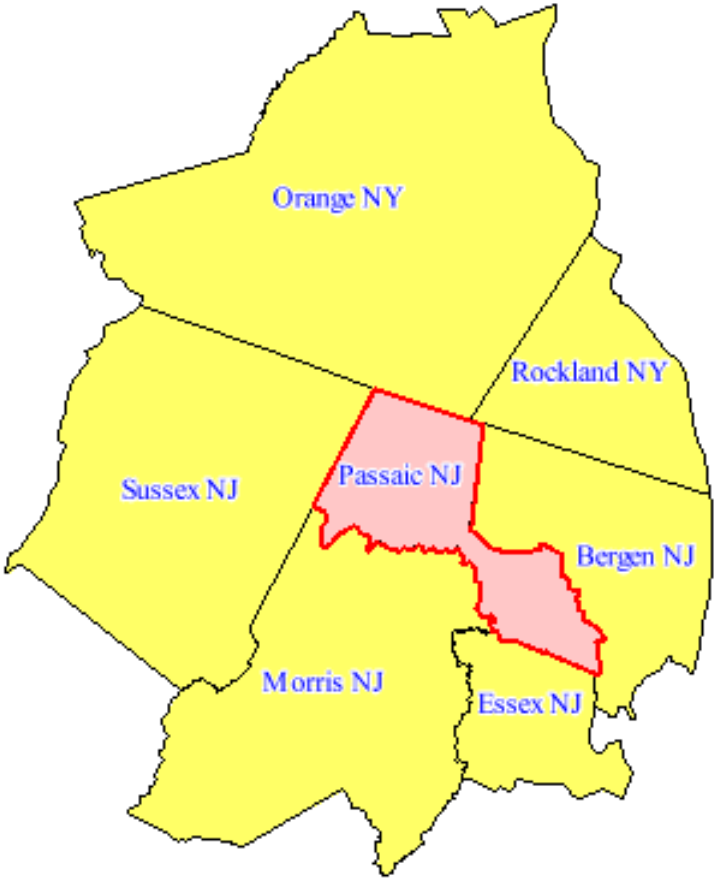
```
SELECT c.county, c.state_abrv
  FROM us_counties c,
       us_states s
 WHERE s.state = 'New Hampshire'
   AND SDO_RELATE (
         c.geom,s.geom,
         'mask=INSIDE+COVEREDBY'
       ) = 'TRUE';
```



Przykład

- **Find all the counties around Passaic County in New Jersey:**

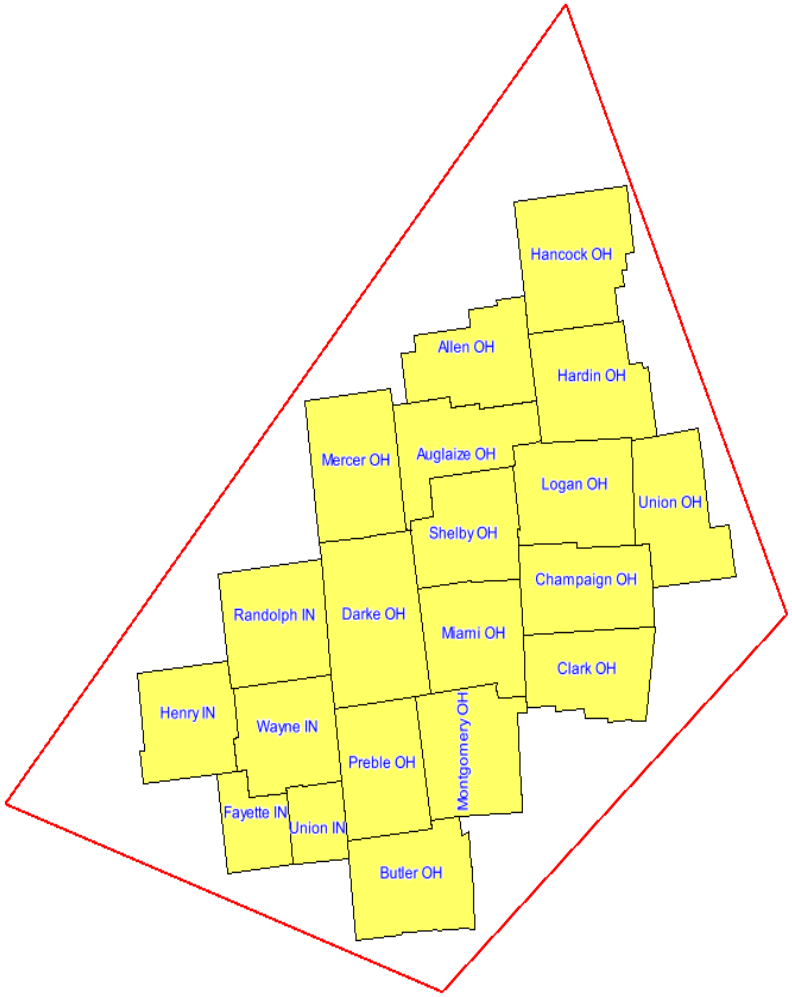
```
SELECT c1.county, c1.state_abrv
  FROM us_counties c1,
       us_counties c2
 WHERE c2.state = 'New Jersey'
   AND c2.county = 'Passaic'
   AND SDO_TOUCH (
         c1.geom, c2.geom
       )='TRUE';
```



Przykład

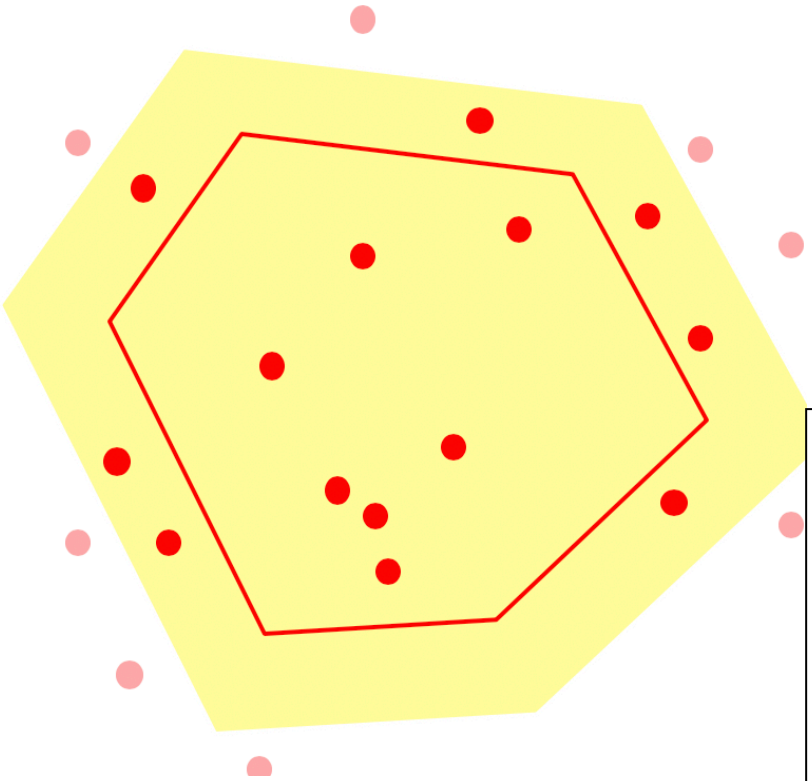
- **Find all counties in a selected area:**

```
SELECT county, state_abrv
  FROM us_counties
 WHERE SDO_INSIDE (
         geom,
         SDO_GEOMETRY(2003, 8307, NULL,
           SDO_ELEM_INFO_ARRAY(1, 1003, 1),
           SDO_ORDINATE_ARRAY(
             -84.505957, 39.139801,
             -82.987381, 39.988933,
             -83.581649, 41.66395,
             -86.145999, 39.7777,
             -84.505957, 39.139801)
         )
       ) ='TRUE';
```



SDO_WITHIN_DISTANCE - przyklad

- How many customers are within 10 km of sales region R1 ?



```
SELECT count(*)
  FROM sales_regions r, customers c
 WHERE r.region_id = 'R1'
   AND sdo_within_distance (
         c.location, r.geom,
         'distance=10 unit=km')
       = 'TRUE';
```

```
SELECT count(*)
  FROM sales_regions r, customers c
 WHERE r.region_id = 'R1'
   AND sdo_within_distance (
         c.location, r.geom,
         'distance=10 unit=km')
       = 'TRUE'
   AND SDO_GEOM.SDO_DISTANCE (
         c.location, r.geom, 0.5) > 0;
```

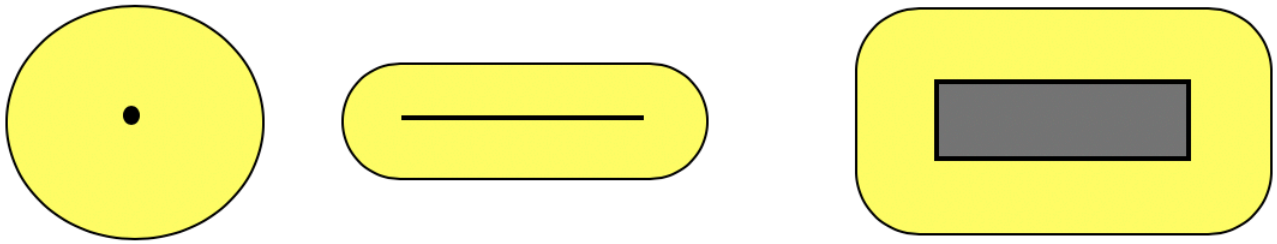
SDO_NN : Przyklad

- Find the five cities nearest to Interstate I170:

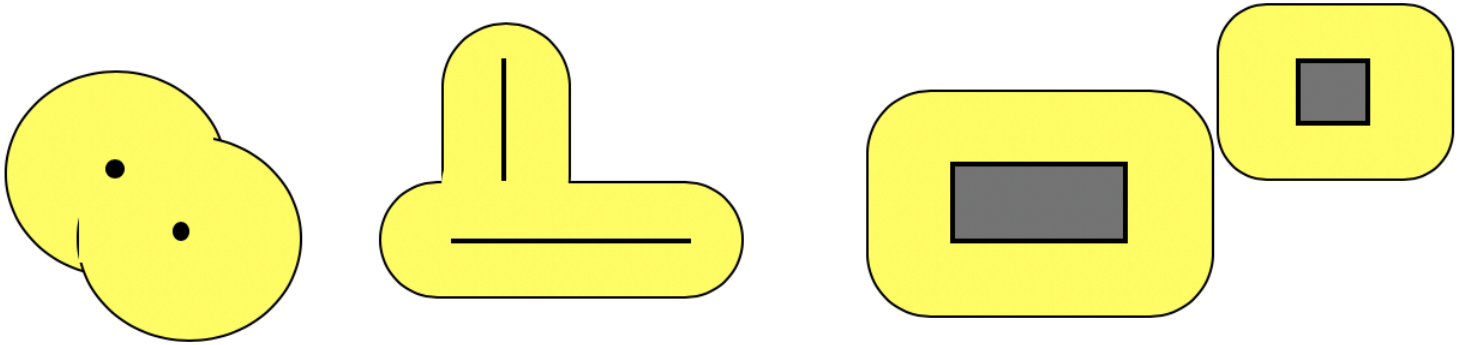
```
SELECT c.city, c.state_abrv
  FROM us_interstates i,
       us_cities c
 WHERE i.interstate = 'I170'
   AND sdo_nn(c.location, i.geom, 'sdo_num_res=5') = 'TRUE';
```

Bufor

- Simple geometries

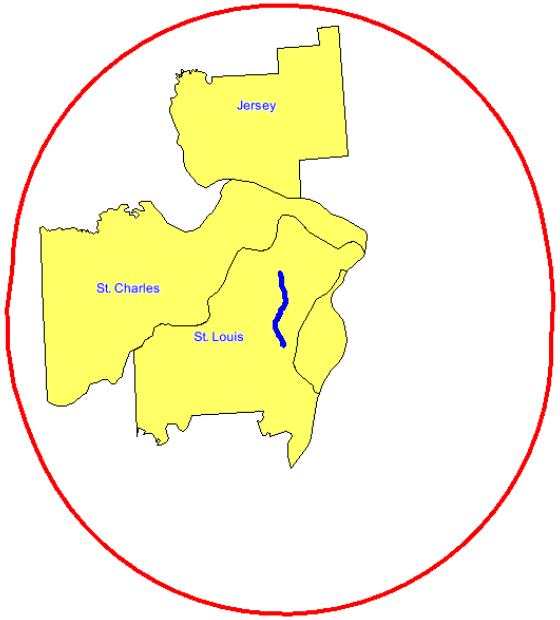


- Collection geometries



Przykład

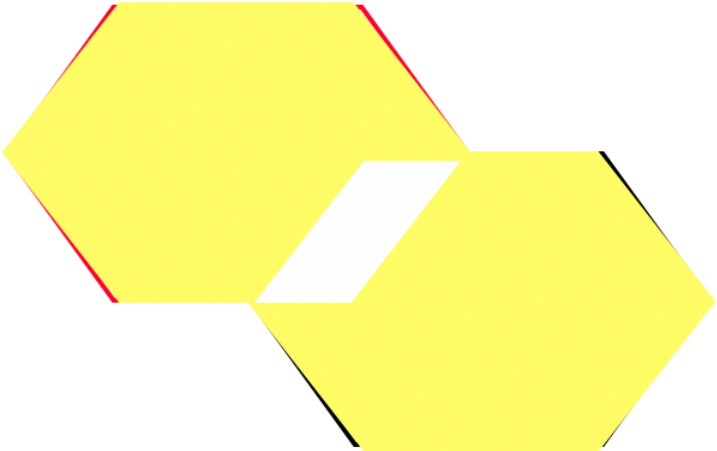
```
SELECT c.state_abrv, c.county
FROM us_interstates i, us_counties c
WHERE i.interstate = 'I170'
  AND sdo_inside (
        c.geom,
        sdo_geom.sdo_buffer (
          i.geom, 60, 0.5,
          'UNIT=KILOMETER')
      ) = 'TRUE';
```



XOR

SDO_GEOM.SDO_XOR

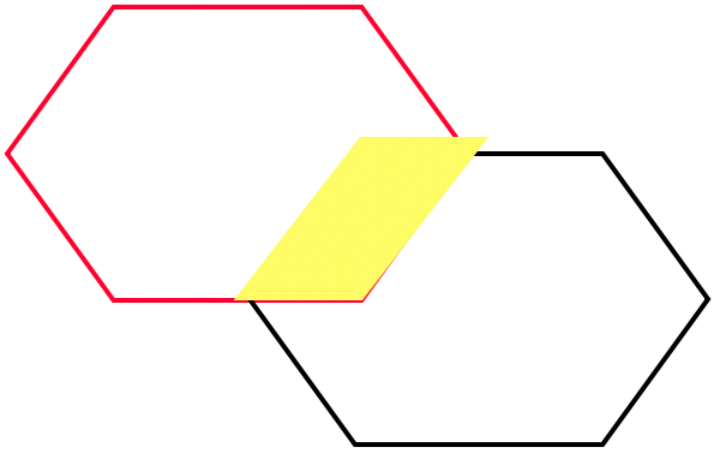
- Generates a geometry representing the symmetric difference between two geometries



Intersection

SDO_GEOM.SDO_INTERSECTION

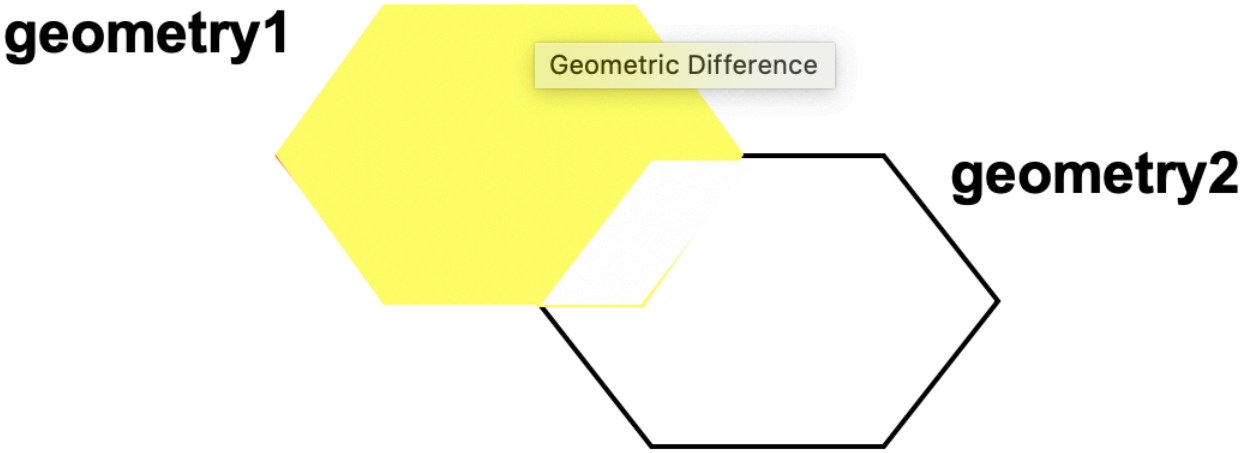
- Generates a geometry representing the intersection of two geometries



DIFF

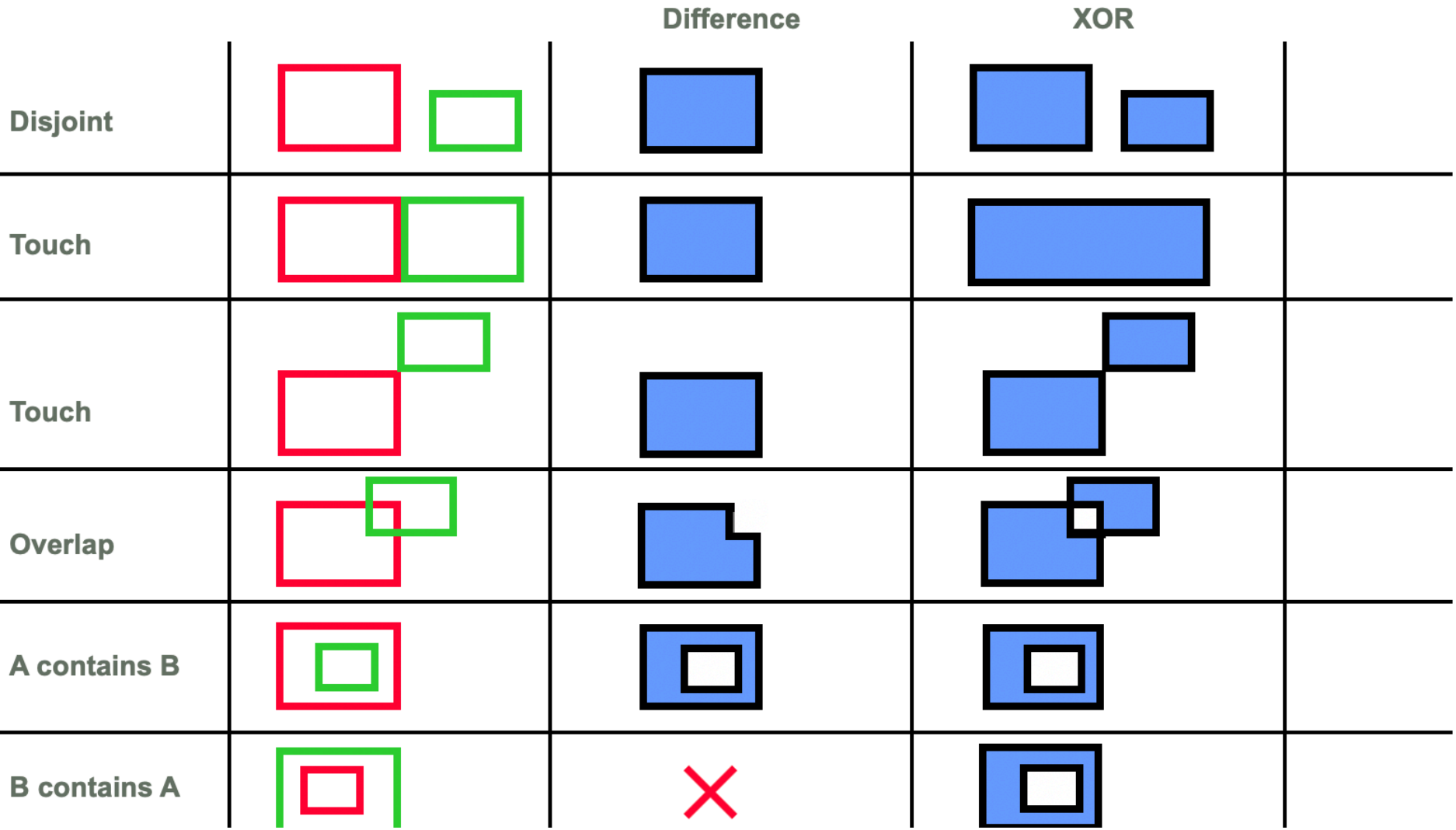
SDO_GEOM.SDO_DIFFERENCE

- Generates a geometry representing the difference between two geometries



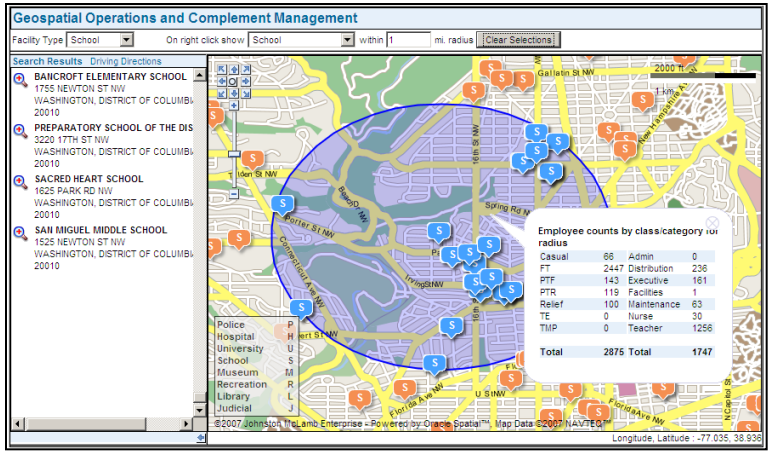
Geometry Operations

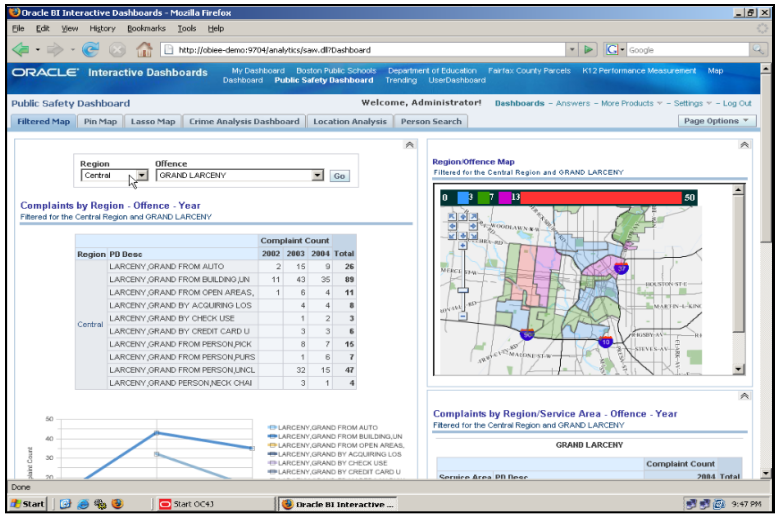
Geometry Operations



- **I wiele innych przydatnych operacji na geometriach**

MapViewer

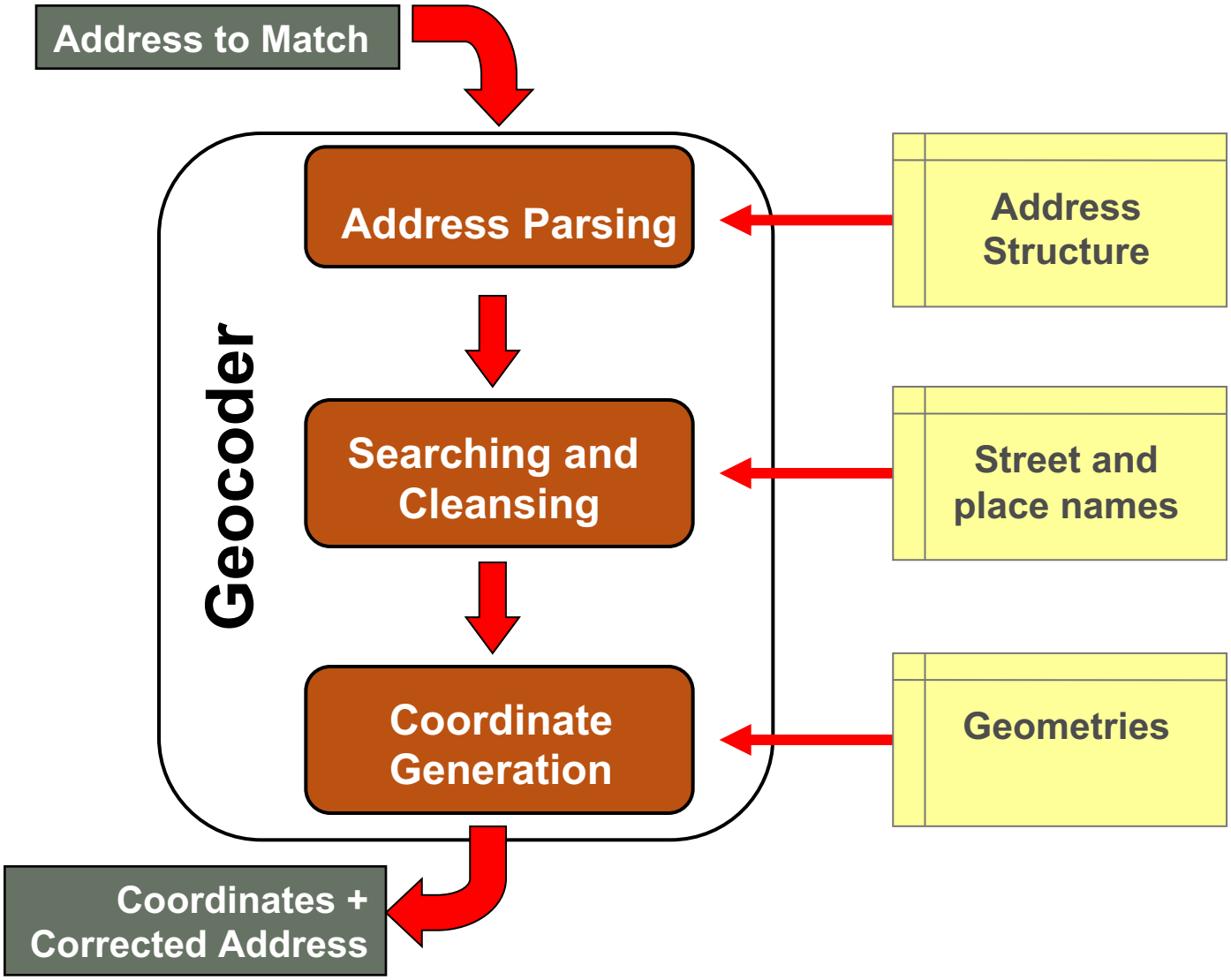

Geospatial Operations and Complement Management
Facility Type School
On right click show School
within 1 mi. radius
Clear Selections
Search Results Driving Directions
BANCROFT ELEMENTARY SCHOOL
1755 NEWTON ST NW
WASHINGTON, DISTRICT OF COLUMBI
20010
PREPARATORY SCHOOL OF THE DIS
3220 17TH ST NW
WASHINGTON, DISTRICT OF COLUMBI
20010
SACRED HEART SCHOOL
1625 PARK RD NW
WASHINGTON, DISTRICT OF COLUMBI
20010
SAN MIGUEL MIDDLE SCHOOL
1526 NEWTON ST NW
WASHINGTON, DISTRICT OF COLUMBI
20010
Employee counts by class/category for radius
Casual 66 Admin 0
FT 2447 Distribution 236
PTF 143 Executive 161
PTR 119 Facilites 1
Relief 100 Maintenance 63
TE 0 Nurse 30
TMP 0 Teacher 1256
Total 2875 Total 1747
Police P
Hospital H
University U
School S
Museum M
Recreation R
Library L
Judicial J
©2007 Johnston, The Lamb Enterprise - Powered by Oracle Spatial
Map Data ©2007 NAVTEQ
Longitude, Latitude : -77.035, 38.936


Oracle BI Interactive Dashboards - Mozilla Firefox
http://obiee-demo:9704/analytics/saw.dll?Dashboard
ORACLE Interactive Dashboards
Public Safety Dashboard
Welcome, Administrator!
Filtered Map
Pin Map
Lasso Map
Crime Analysis Dashboard
Location Analysis
Person Search
Region Central
Offence GRAND LARCENY
Go
Complaints by Region - Offence - Year
Filtered for the Central Region and GRAND LARCENY
Region/Offence Map
Filtered for the Central Region and GRAND LARCENY
Complaints by Region/Service Area - Offence - Year
Filtered for the Central Region and GRAND LARCENY
GRAND LARCENY

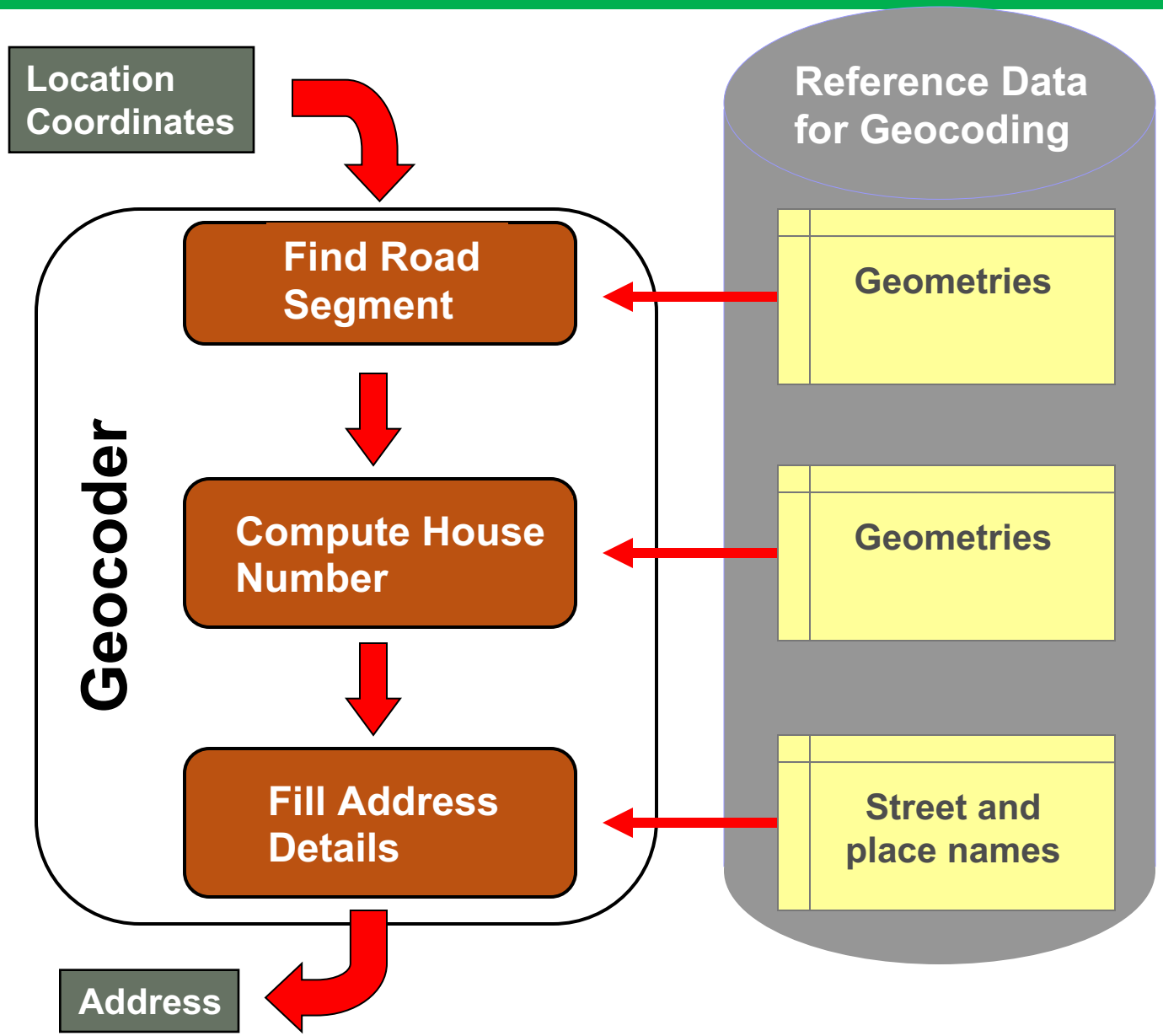
Geokodowanie

- **Zamiana adresu na współrzędne**
- **Zamiana współrzędnych na adres**

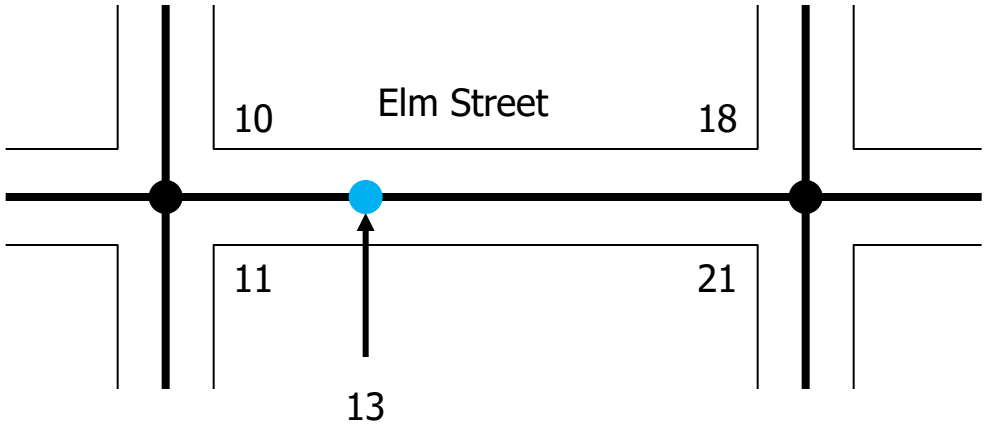
Geokodowanie

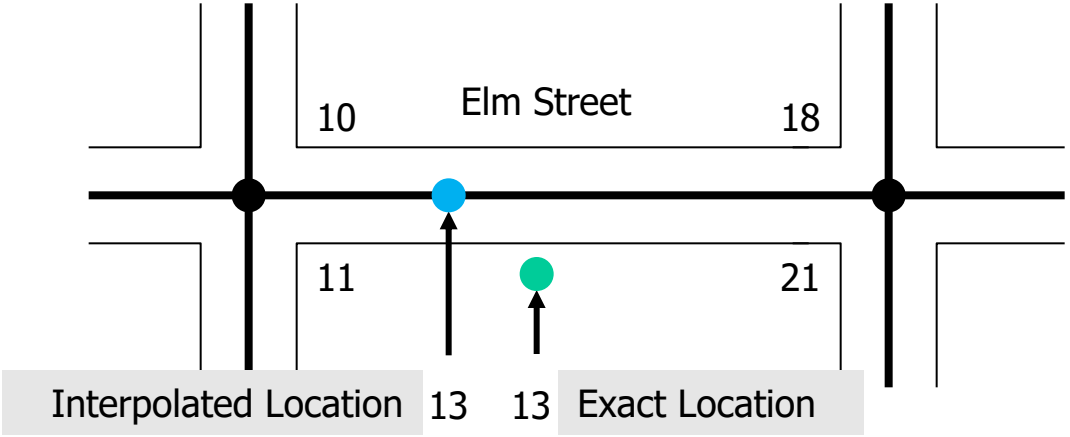


Geokodowanie odwrotne



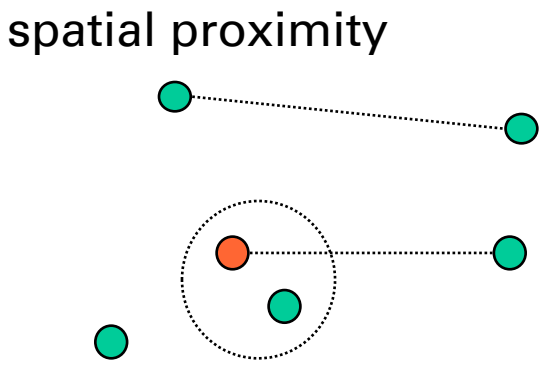
Aproksymacja

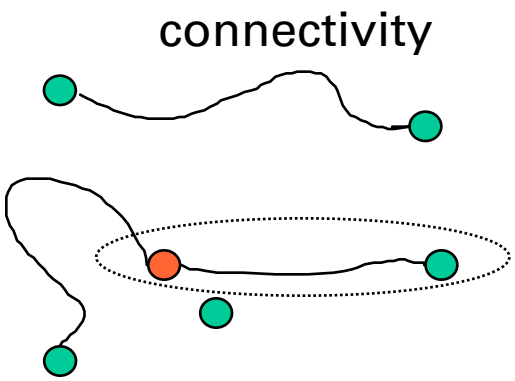




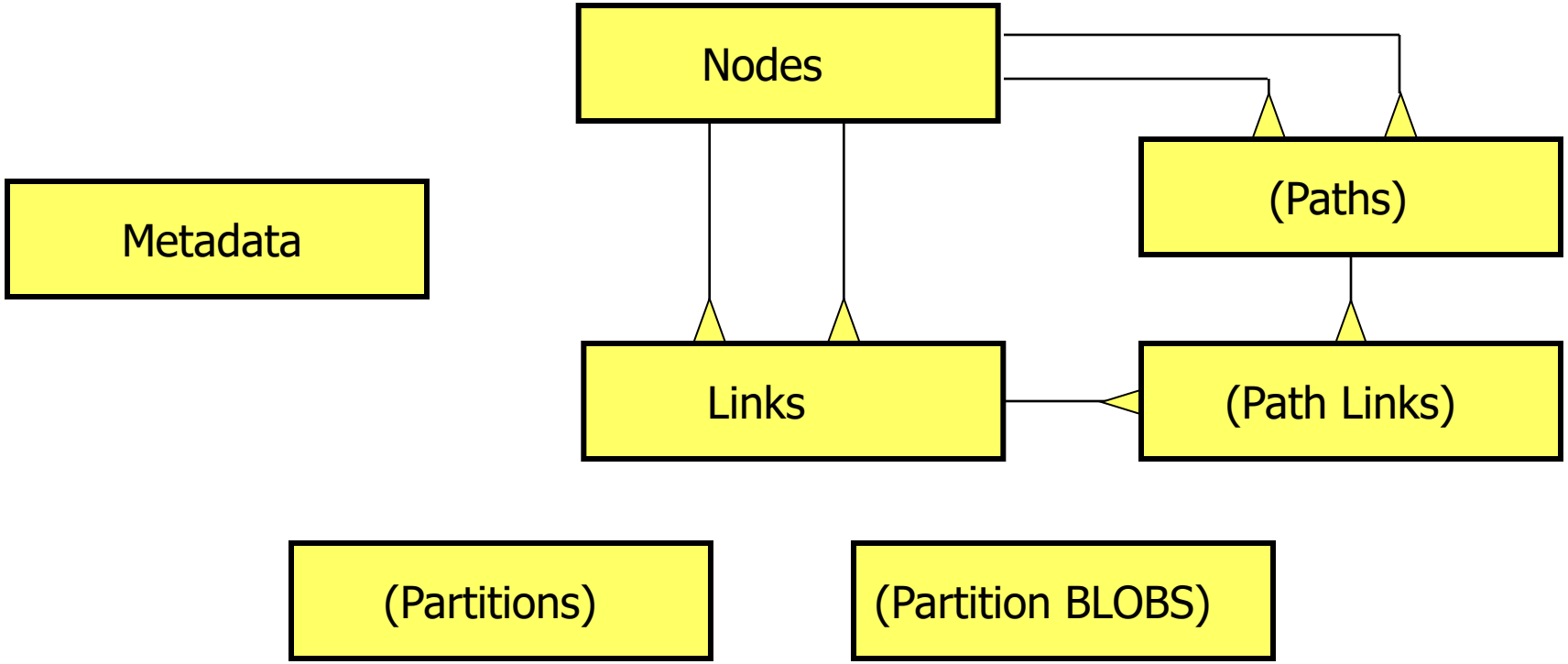
Analizy sieciowe

- **Analizy przestrzenne vs analizy sieciowe**





Struktury danych



Struktury danych

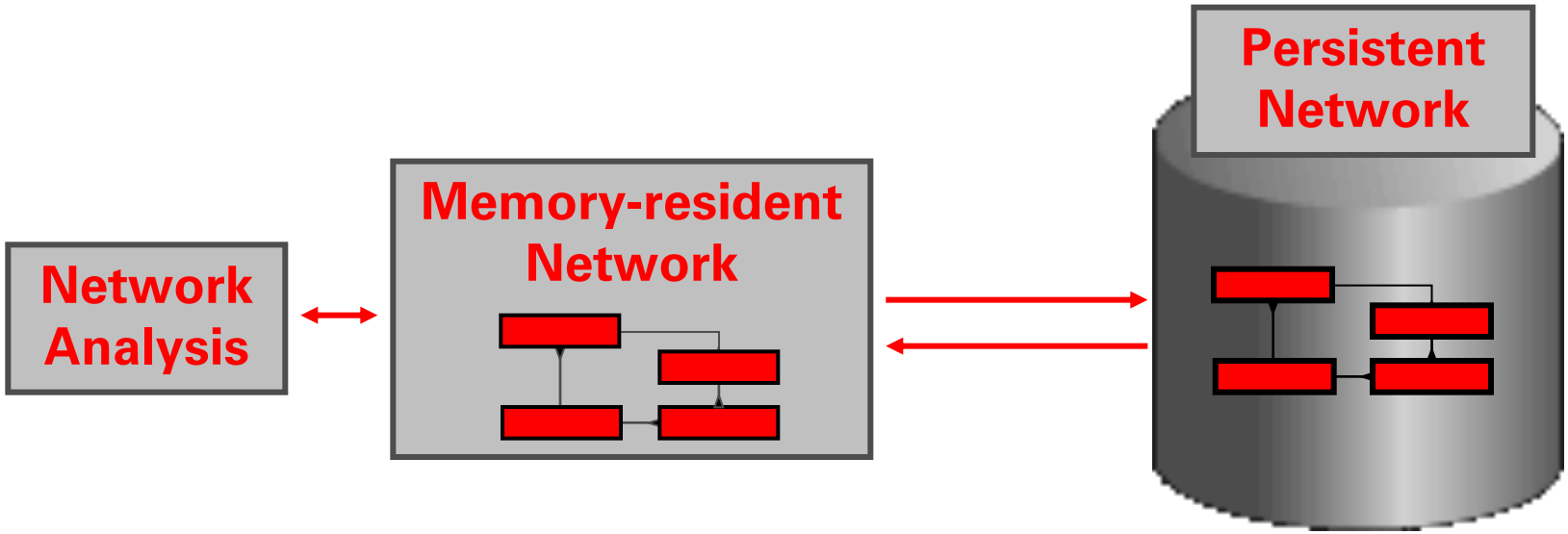
Nodes

Node ID
N1
N2
N3
N4
N5
N6
N7
N8
N9

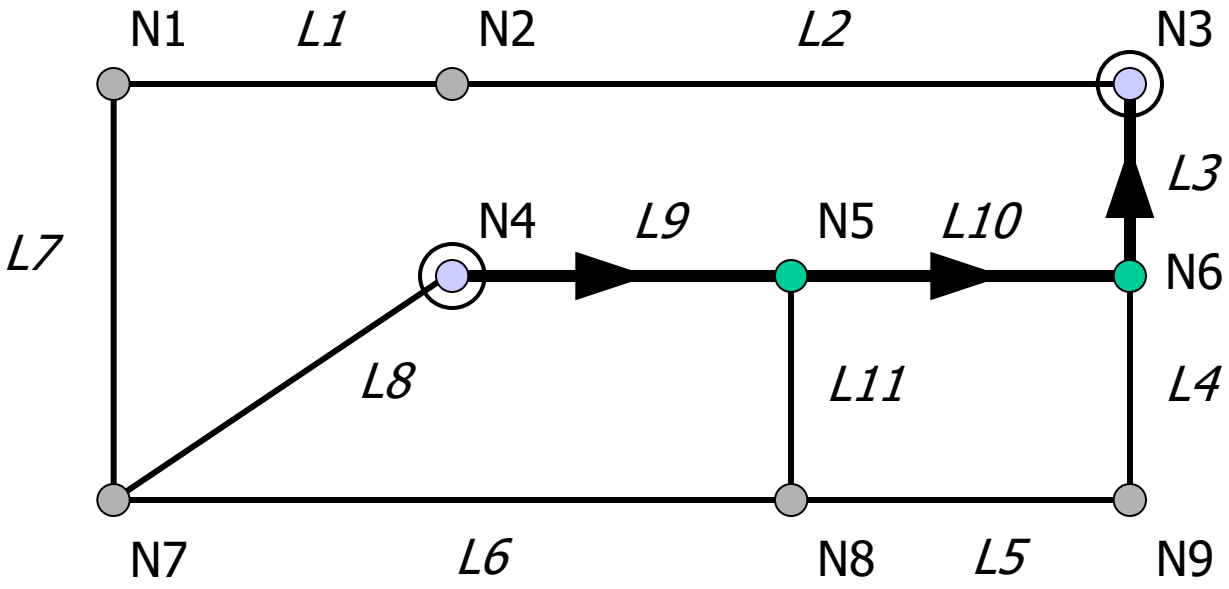
Links

Link Id	Start Node	End Node	Cost	Directed
L1	N2	N1	1	YES
L2	N3	N2	2	YES
L3	N6	N3	1	NO
L4	N6	N9	1	NO
L5	N9	N8	1	NO
L6	N8	N7	3	YES
-L6	N7	N8	2	YES
L7	N1	N7	2	YES
L8	N4	N7	1.5	YES
L9	N5	N4	1	YES
L10	N5	N6	1	YES

Struktury danych



Model sieciowy



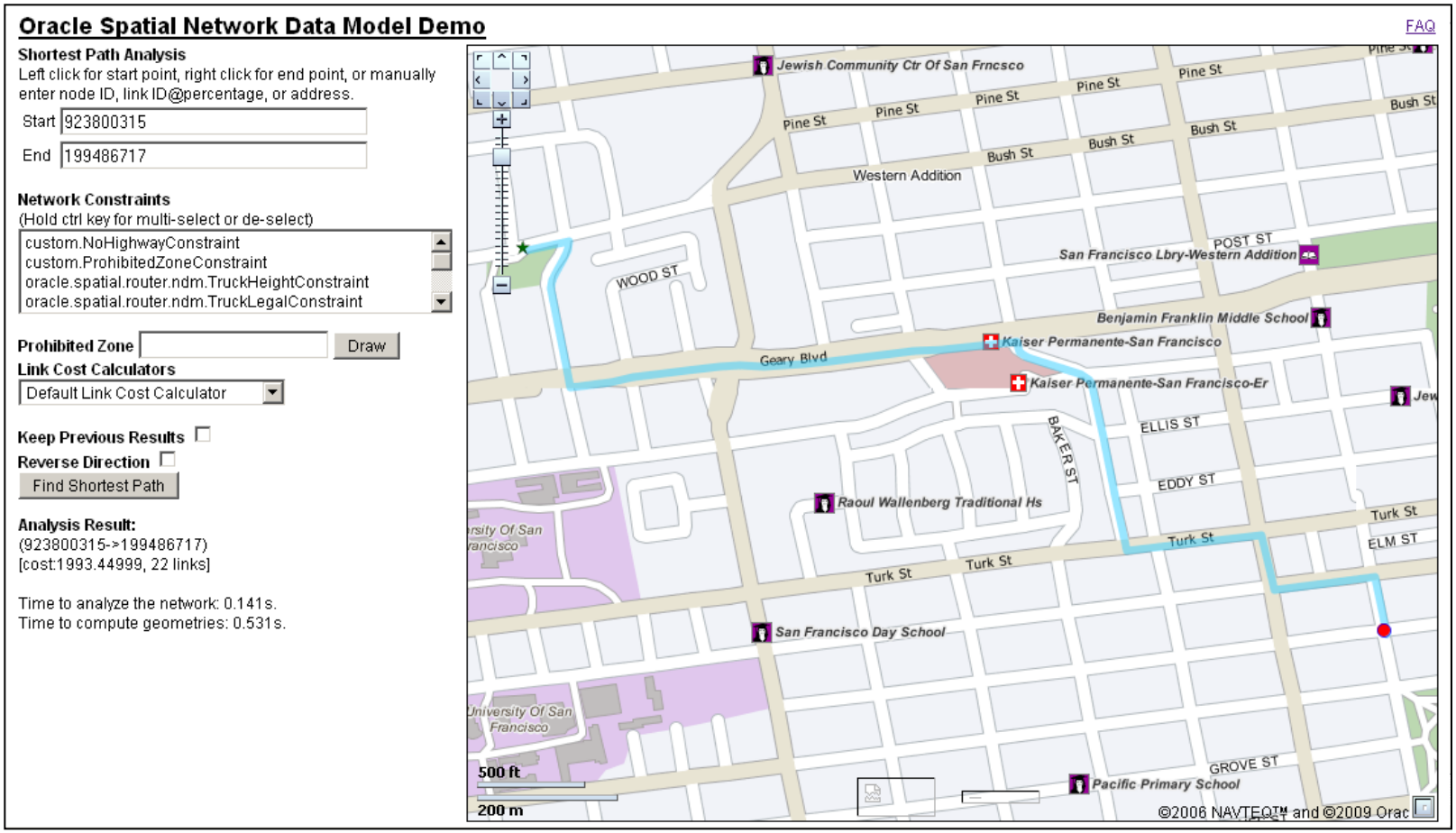
Shortest path from node N4 to node N3

```
LogicalSubPath subPath =
  analyst.shortestPathDijkstra (new PointOnNet(startNodeId),
    new PointOnNet(endNodeId));
```

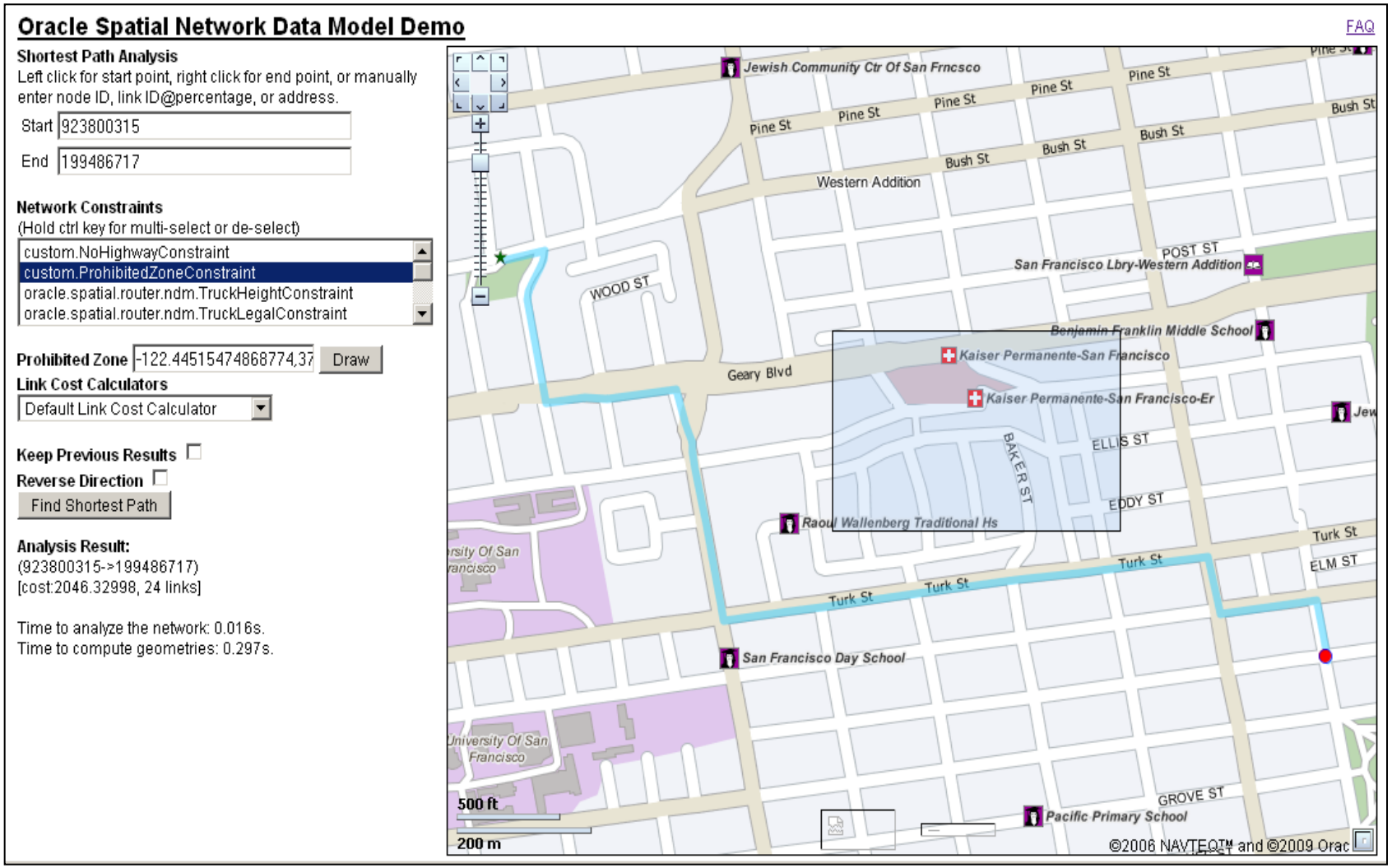
Przykłady operacji

- **shortestPathDijkstra() and shortestPathAStar()**
  - Find the shortest path between two nodes
- **nearestNeighbors() and nearestReachingNeighbors()**
  - Find the N nearest nodes from a node
- **withinCost() and withinReachingCost()**
  - Find all nodes within a chosen distance (=cost) from a node
- **findReachableNodes()**
  - Find all nodes that can be reached from a node
- **findReachingNodes()**
  - Find all nodes capable of reaching a node
- **findConnectedNodes()**
  - Finds all nodes connected to a node
- **isReachable()**
  - Find whether a path exists between two nodes
- **traceIn()**
  - Finds the nodes and partial links within a specified cost to a target node
- **traceOut()**
  - Finds the nodes and partial links within the specified cost from a source node

Najkrótsza ścieżka



Wyłączenie obszary



Obszar do którego można dotrzeć w zadanym czasie

Oracle Spatial Network Data Model Demo

FAQ

Within Cost Polygon Analysis

Start 199362936

Cost 2000

Link Cost Calculators

Default Link Cost Calculator

Keep Previous Results ☑

Reverse Direction ☐

Show **Concave** Polygon Within the Given Cost

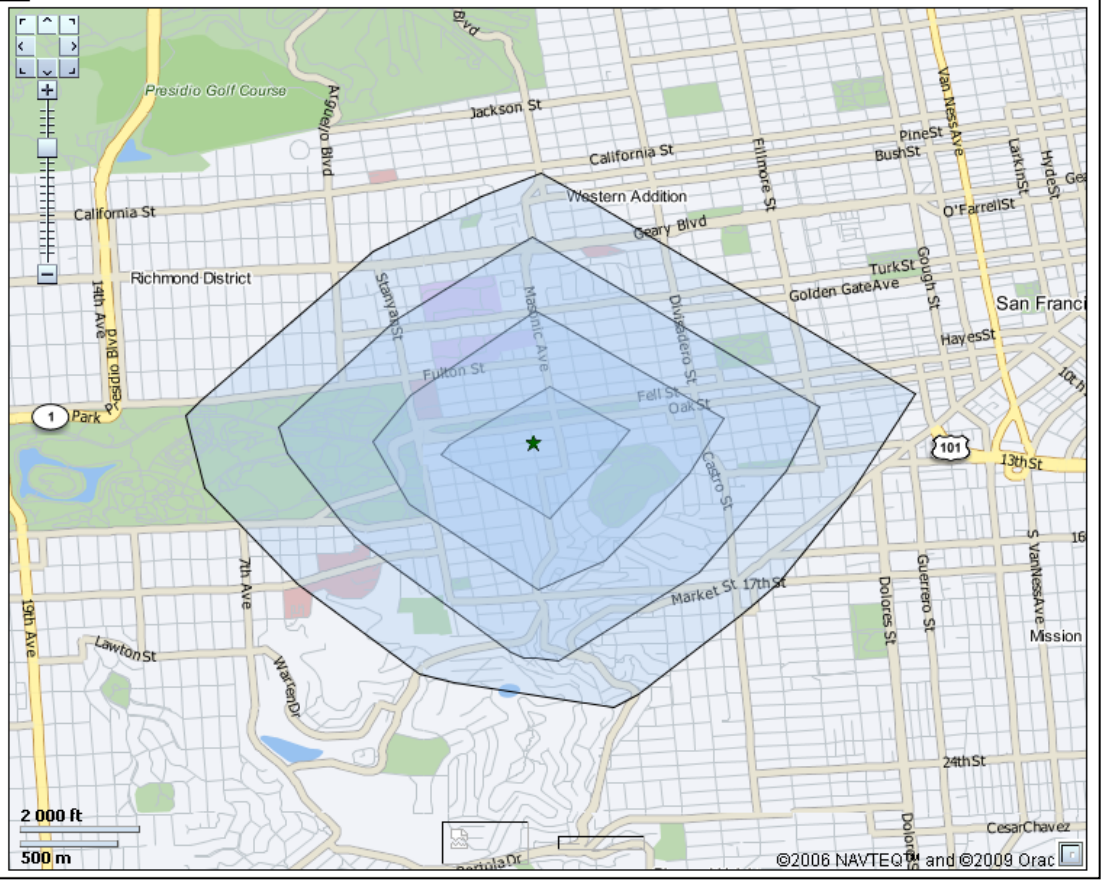
Show **Convex** Polygon Within the Given Cost

Analysis Result:

JGeometry (gtype=3, dim=2, srid=8307,
ElemInfo(1,1003,1),
Ordinates(-122.446226297628,37.7664793800805
-122.445524992341,37.7672057684043
-122.443033233211,37.7697984010583
-122.441501814778,37.7716649611386
-122.442290674963,37.772124829214
-122.446187118027,37.7743341588066
-122.452149294084,37.7708473562175
-122.452728423596,37.7702246892484
-122.451391797772,37.7694280073557
-122.450622062112,37.7689873533831
-122.447160769367,37.7670115506401
-122.446226297628,37.7664793800805
))
Time to analyze the network: 1.797s.

Analysis Result:

JGeometry (gtype=3, dim=2, srid=8307,
ElemInfo(1,1003,1),
Ordinates(-122.446904565434,37.7622265396092
-122.442936657,37.7639331689957
-122.441495406314,37.76527
-122.437880475036,37.7690723259629
-122.435887585414,37.7723808527655



Koniec