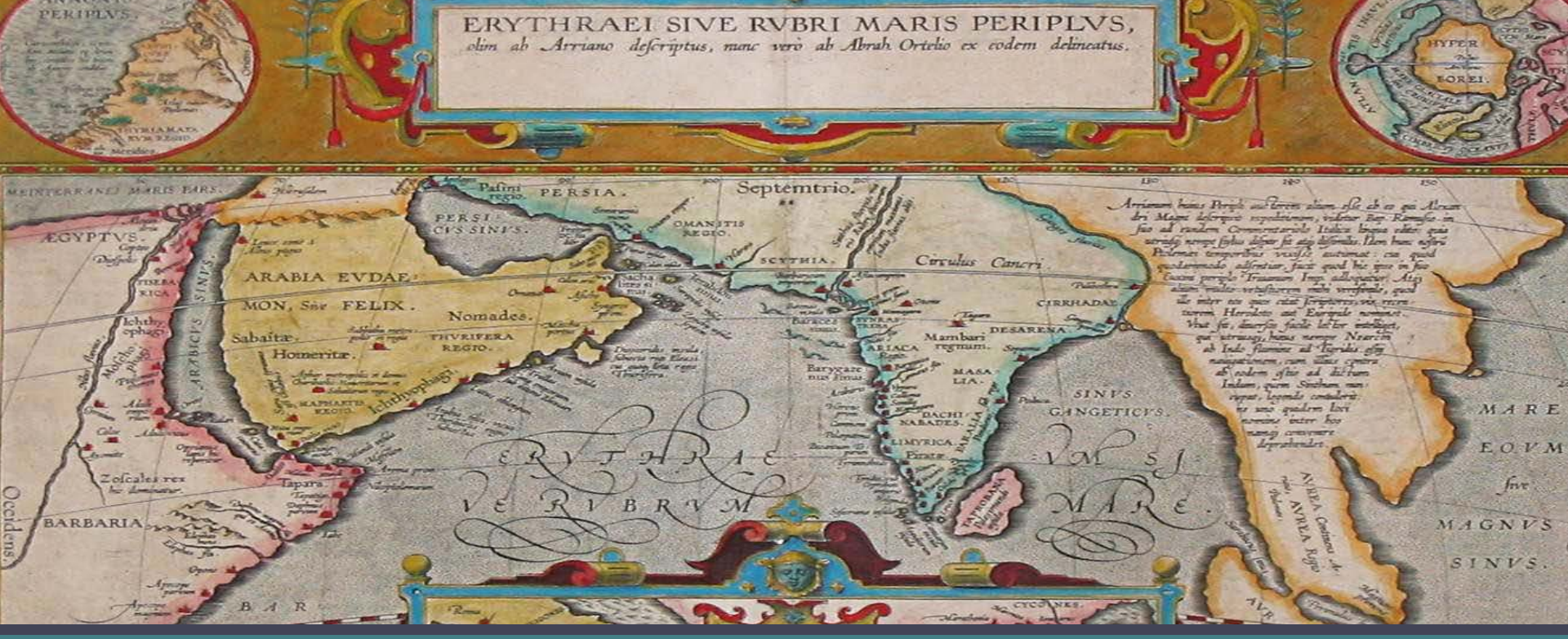
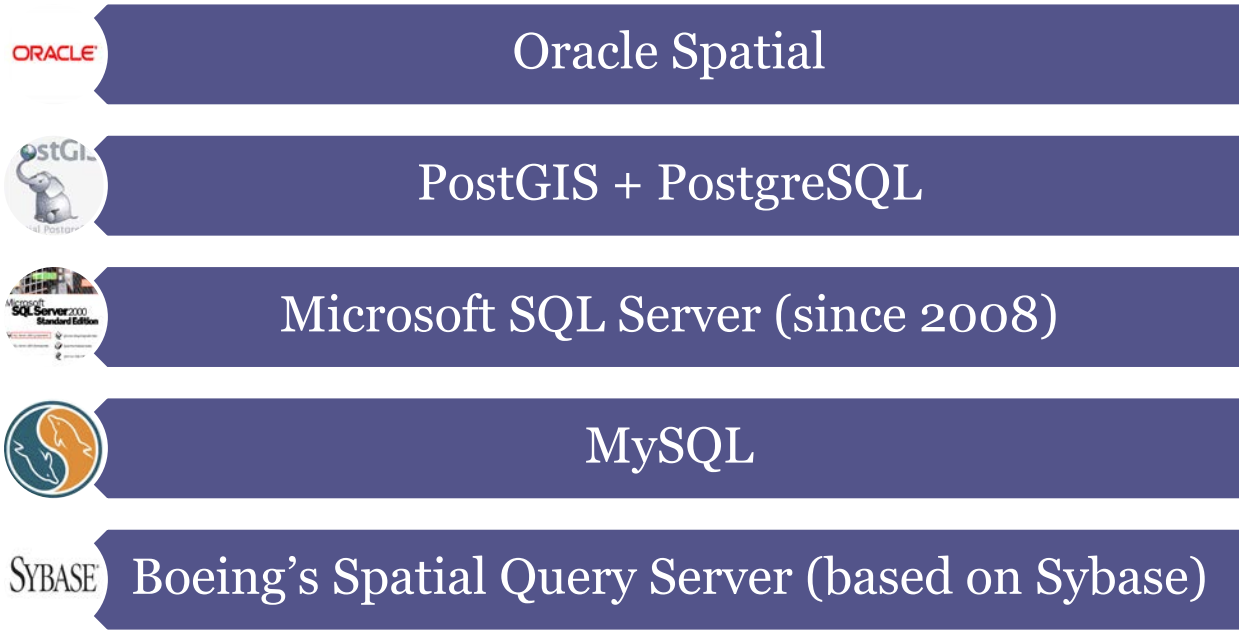


Spatial Databases

Spatial Databases

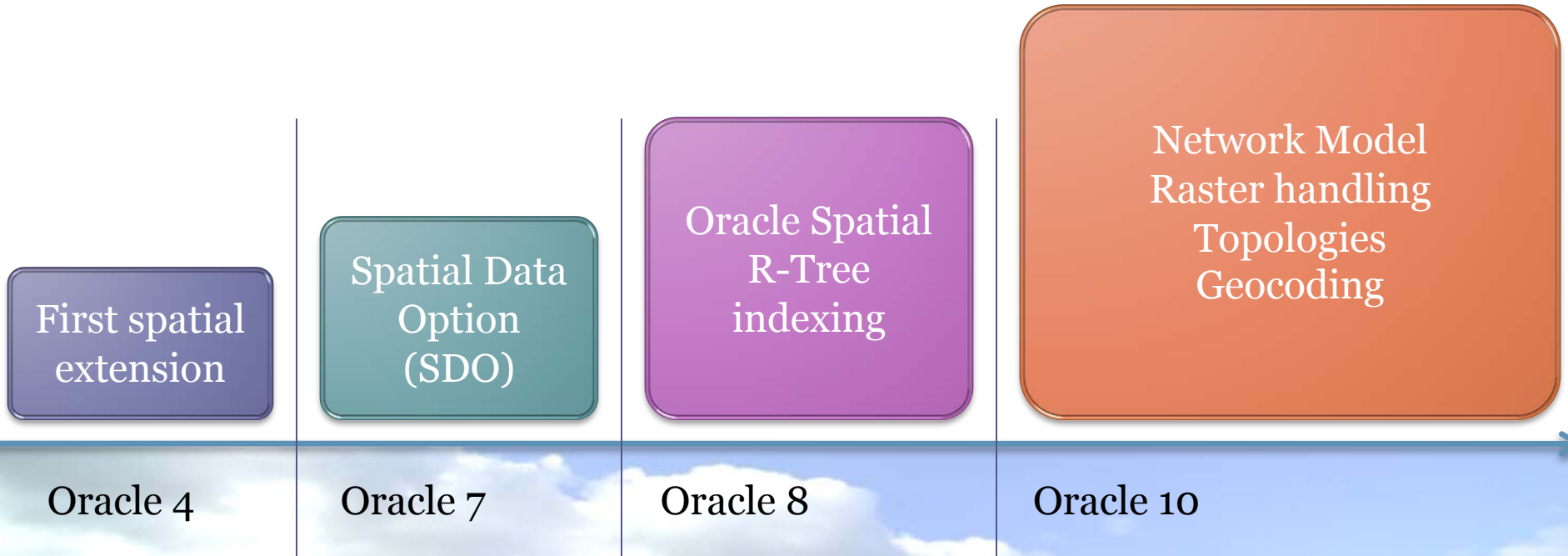
- **A spatial database** is a database that is optimized to store and query data related to objects in space, including points, lines and polygons.

- Up to now the market proposes:

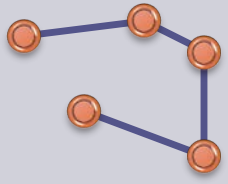
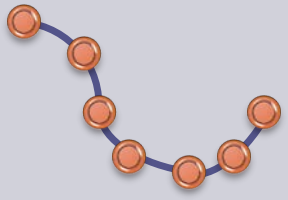
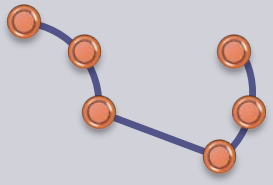
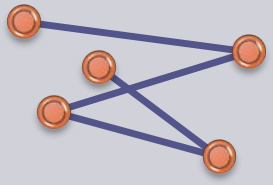


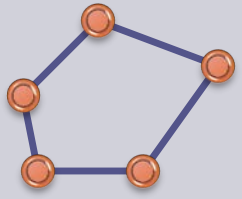
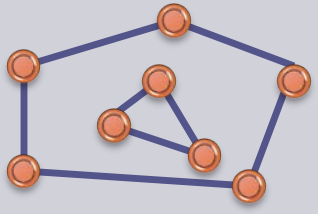
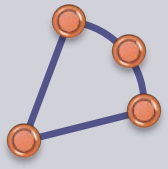
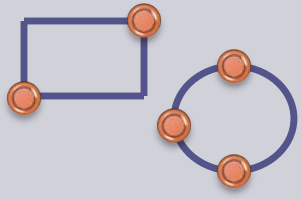
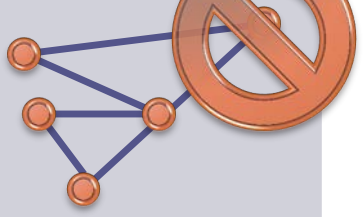
Oracle Spatial Extensions

- Forms a separately-licensed option component of the Oracle Database to aid users in managing geographic and location-data natively

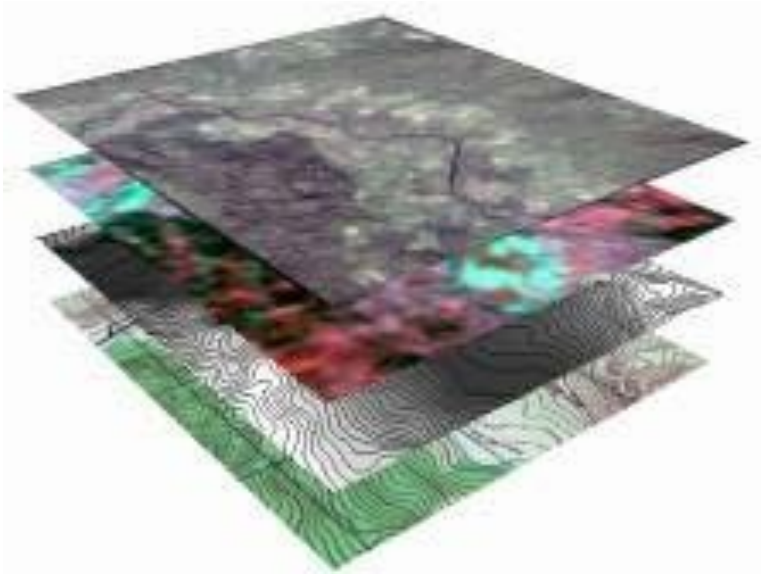


Supported geometries

Point	Line	Arcs	Composed	Intersected
				

Polygon	Holes	Composed	Optimized	Intersected
				

Spatial Database Generalities

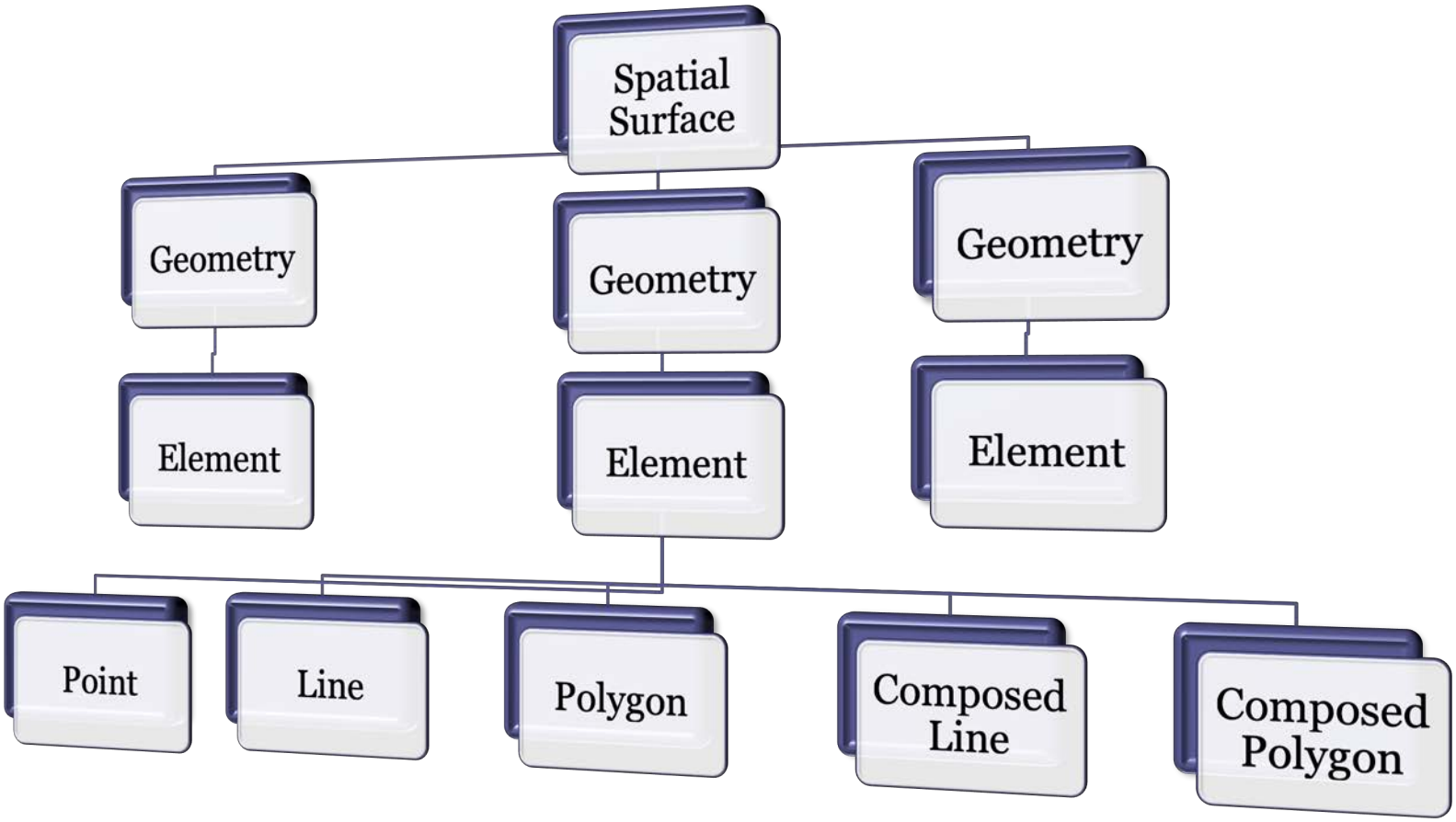


Supported Data Formats

- While managing spatial data you might have to import data from heterogeneous sources
- Different GIS (Geographic Information Systems) softwares might support different data types
- Visit http://www.safe.com/products/desktop/formats/index.php for a complete list data sources – supported types

Oracle Hierarchical Model

- Oracle datatypes have a hierarchical model



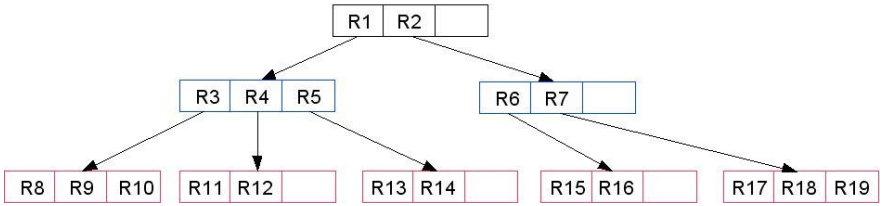
Geometries

- Geometries refers to spatial objects
- Their distribution is not uniform but if follows the spatial location of real entities

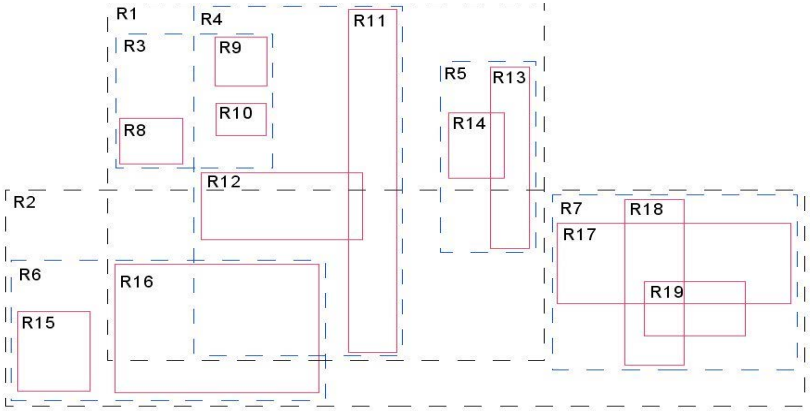
Spatial Indexing

- Special indexes are needed to face performant storage and query of geometries

- Most famous are
  - R-Tree
  - Quad-Tree

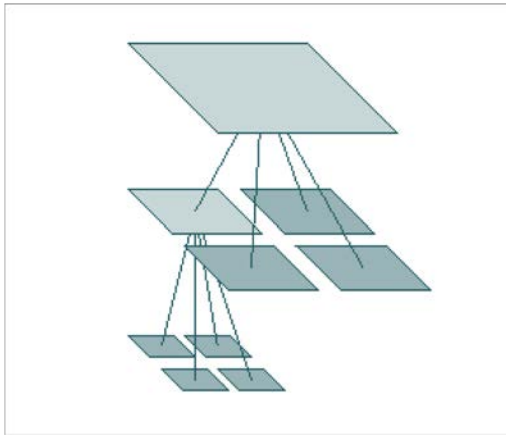


R - Tree



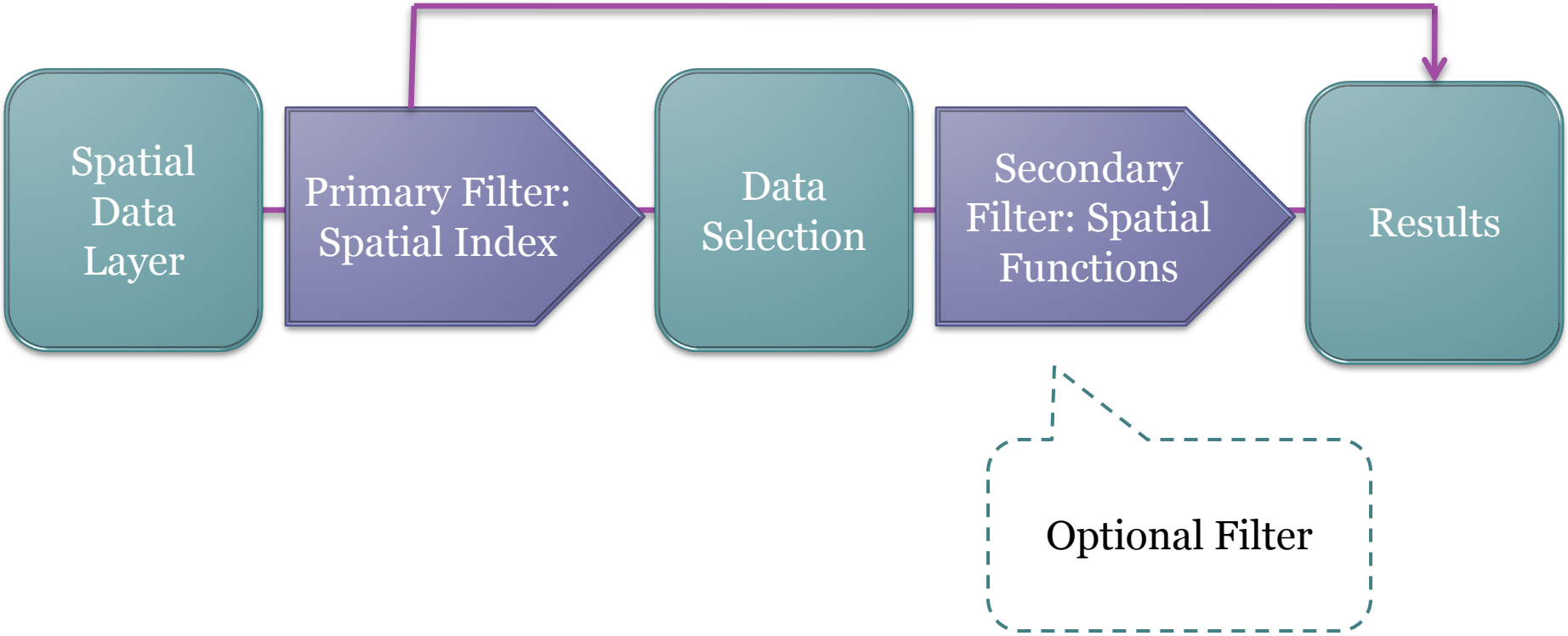
- **R-trees** are tree data structures that are similar to B-trees, but are used for indexing multi-dimensional information;
- A common real-world usage for an R-tree might be: "Find all museums within 2 kilometres (1.2 mi) of my current location".
- The data structure splits space with hierarchically nested, and possibly overlapping, minimum bounding rectangles
- (MBRs, otherwise known as bounding boxes, i.e. "rectangle", what the "R" in R-tree stands for).

Quadtree

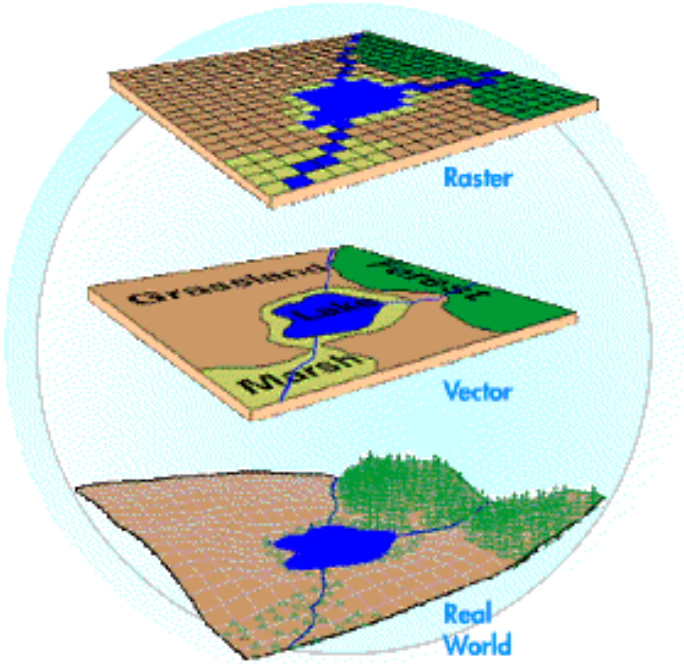


- A **quadtree** is a tree data structure in which each internal node has up to four children.
- Quadtrees are most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions.
- The regions may be square or rectangular, or may have arbitrary shapes.

Spatial Query Processing



Oracle Spatial Data Types



SDO_GEOMETRY

- Is composed of

Methods

GET_DIMS()	:NUMBER
GET_GTYPE	:NUMBER

SDO_GTYPE	NUMBER
SDO_SRID	NUMBER
SDO_POINT	SDO_POINT_TYPE
SDO_ELEM_INFO	SDO_ELEM_INFO_ARRAY
SDO_ORDINATES	SDO_ORDINATE_ARRAY

```
CREATE TABLE forest(
        name                VARCHAR2(30),
        animal_population   NUMBER(9),
        geom                MDSYS.SDO_GEOMETRY);
```

SDO_GTYPE

- It defines the geometry type stored into the SDO_GEOMETRY object

Dimensions

Geometry Type

Name	2D	3D	4D
UNKNOWN_GEOMETRY	2000	3000	4000
POINT	2001	3001	4001
LINESTRING	2002	3002	4002
POLYGON	2003	3003	4003
COLLECTION	2004	3004	4004
MULTIPOINT	2005	3005	4005
MULTILINESTRING	2006	3006	4006
MULTIPOLYGON	2007	3007	4007

NOT OGC

SDO_POINT

Example:
GPS Receiver

- Used to store Point data
- It is ignored if SDO_ELEM_INFO and SDO_ORDINATES are not NULL
- It is of type **SDO_POINT_TYPE**

X	NUMBER
Y	NUMBER
Z	NUMBER

SDO_ELEM_INFO

- It is store as a SDO_ELEM_INFO_ARRAY:

  VARRAY (1048576) OF NUMBER

- Its entries must be considered grouping elements by 3
  - Starting position of the coordinates (starting from 1)
  - Element type (as shown in the table)
  - Interpretation

Number	Element Type	Interpretation
0	UNKNOWN_ELEMENT	
1	POINT	Number of points of the collection
2	LINESTRING	1. Lines
3	POLYGON	1 lines
1003	External	2 arcs
2003	Internal	3 optimized rectangle 4 circle
4	COMPOUND LINESTRING	Number of elements
5	COMPOUNG POLYGON	Number of elements
1005	External	
2005	Internal	

SDO_ORDINATES

- Stored as SDO_ORDINATE_ARRAY

  VARRAY (1048576) OF NUMBER

- It contains a list of object’s coordinates

Examples 1/5 – inserting a Point

```
INSERT INTO pollution VALUES (
34.6,57.4,etc…,
        MDSYS.SDO_GEOMETRY(
        3001, 352257,
    MDSYS.SDO_POINT_TYPE(521030,240120,550),
        null,null)
);
```

SDO_GTYPE

SDO_SRID

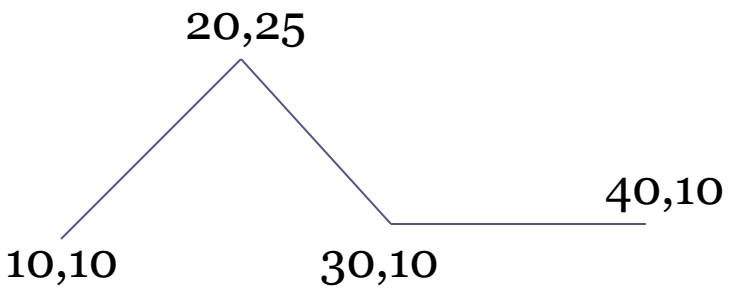
SDO_POINT

SDO_ELEM_INFO

SDO_ORDINATES

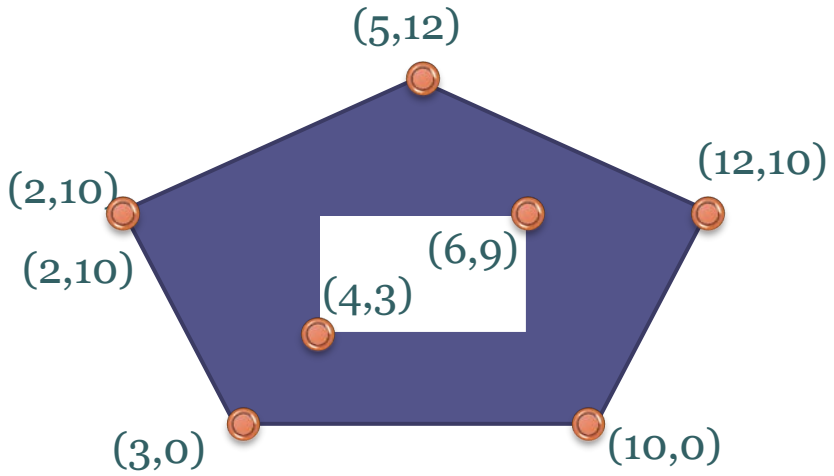
Examples 2/5 – inserting a line

```
INSERT INTO roads VALUES ('motorway','NYC',etc…,
   mdsys.sdo_geometry(
   2002,352257,null,
         mdsys.sdo_elem_info_array(1,2,1),
         mdsys.sdo_ordinate_array(10,10, 20,25, 30,10, 40,10))
);
```

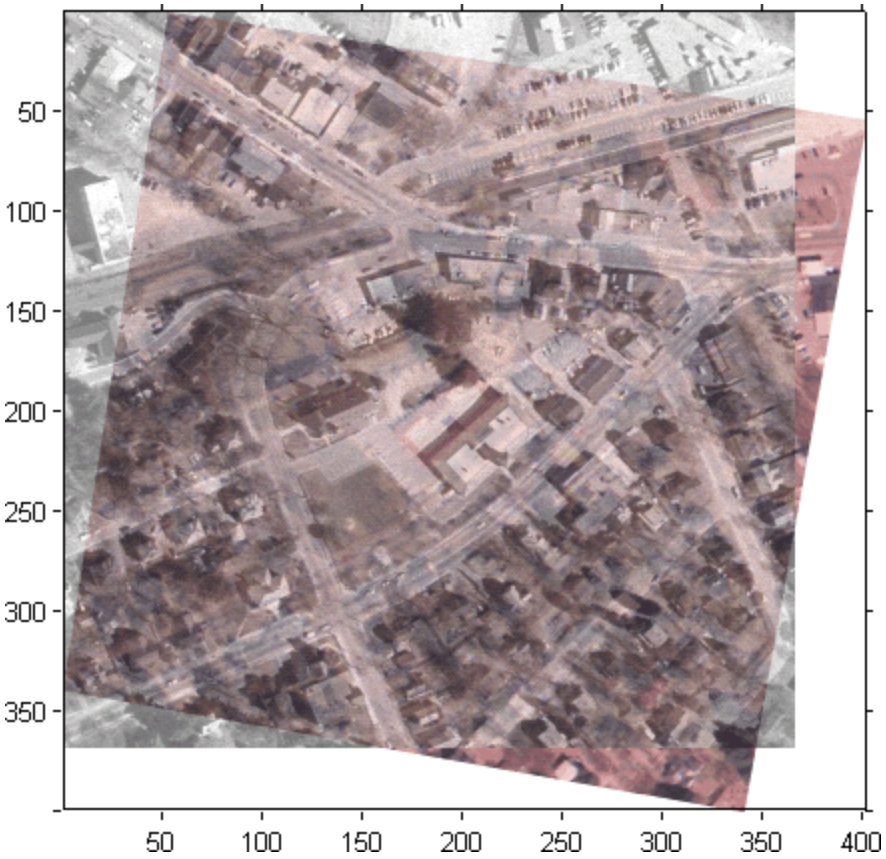


Examples 3/5 – inserting a polygon

```
INSERT INTO roads VALUES ('motorway','NYC',etc…,
   mdsys.sdo_geometry(
        2003,352257,null,
        mdsys.sdo_elem_info_array(1,1003,1, 13,2003,3),
        mdsys.sdo_ordinate_array(2,10, 3,0, 10,0, 12,10,
        5,12, 2,10, 4,3, 6,9))
);
```

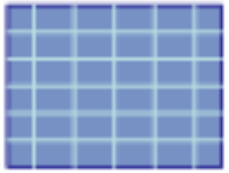


Oracle Spatial Coordinates System

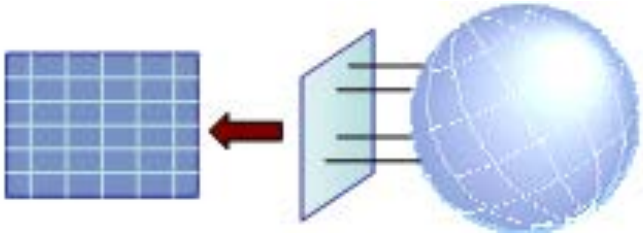


Projections

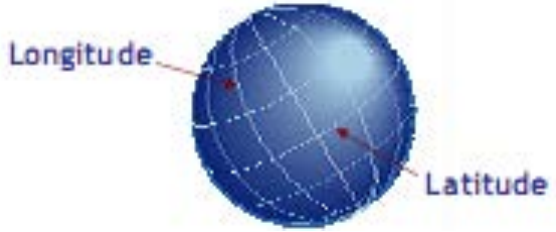
- Projected and Georeferenced



  - 2D projection of the global sphere
  - Cartesian projections
    - Example : Swiss projection



- Not Georeferenced
  - Local systems not linked to the earth global representation
  - Cartesian projections
    - Ex ample: «non-earth» of MapInfo GIS software
- Georeferenced not projected



  - Not projected on a specific plane
  - Latitude Longitude positioning system
    - Example: Oracle, decimal degrees –180 to 180 and – 90 to 90
    - Example : WGS84

SDO_SRID

- It is a INTEGER number that identifies which coordinate system is used by the SDO_GEOMETRY

- The table MDSYS.CS_SRS is defining such reference systems.

Description	Type	Description
CS_NAME	VARCHAR2(68)	Short Name
SRID	NUMBER(38)	Reference System ID
AUTH_SRID	NUMBER(38)	'Oracle'
AUTH_NAME	VARCHAR2(256)	SRID name
WKTEXT	VARCHAR2(2046)	OGC name
CS_BOUNDS	MDSYS.SDO_GEOMETRY	Can be null

How to Change the SRID?

1. Drop Spatial Indexes

```
drop index town_sidx;
```

2. Update metadata table

```
update user_sdo_geom_metadata set srid = 352257
where table_name = 'towns' and column_name = 'geom';
```

3. Update the SDO_SRID of each object

```
update towns t set t.geom.sdo_srid = 352257;
```

4. Adjust tolerances

```
update user_sdo_geom_metadata u set u.diminfo =
mdsys.sdo_dim_array(
        mdsys.sdo_dim_element('x',400000,600000,0.005),
        mdsys.sdo_dim_element('y',100000,300000,0.005)),
where table_name = 'towns' and column_name = 'geom'
```

5. Recreate Spatial Indexes

```
create index  town_idx on towns(geom)
indextype is mdsys.spatial_index;
```

Oracle Spatial Operators

Spatial Operators

- Make use of spatial indexes
- They MUST have a spatial index defined at least for the first spatial type used
- They are only used within the WHERE clause

SDO_FILTER

SDO_FILTER(<geometry-1>,<geometry-2>, 'QUERYTYPE=WINDOW') = 'TRUE' ;

- geometry-1 (searched object)
  - Must be a column table
  - Must be a SDO_GEOMETRY
  - Must be indexed
- geometry-2
  - Must be a variable or column table
  - Must be SDO_GEOMETRY
- 'QUERYTYPE = WINDOW'
  - Necessary parameter
  - Oracle advises to use only WINDOW

SQL EXAMPLE

```
select s.nom, s.type from sondages s
Where sdo_filter ( s.geom,
mdsys.sdo_geometry (2003,352257,null,
mdsys.sdo_elem_info_array(1,1003,3),
mdsys.sdo_ordinate_array(540300,212500,541300,213500)),
'querytype=WINDOW') = 'TRUE';
```

SDO_RELATE

SDO_RELATE(<geometry-1>,<geometry-2>, 'MASK=<mask> QUERYTYPE=WINDOW') = 'TRUE' ;

- Used similarly to SDO_FILTER
- You can use MASK keywords linked by «+» that means OR operation
- For example: COVEREDBY, TOUCH, INSIDE, ANYINTERACT

SQL EXAMPLE

```
select t1.name, t1.na,e
from towns t1, towns t2
where c2.name = 'NYC' and
sdo_relate ( t1.geom,t2.geom,
'mask=TOUCH querytype=WINDOW')='TRUE';
```

SDO_WITHIN_DISTANCE

SDO_WITHIN_DISTANCE(<geometry-1>,<geometry-2>, 'DISTANCE=<n>,[optional parameters]') = 'TRUE';

- geometry-2
  - A buffer will be created starting from this object
- DISTANCE
  - Distance to be considered
- QUERYTYPE (optional)
  - Using a primary and secondaryfilter
  - Si QUERYTYPE=FILTER: uses a primary filter only
- UNIT (optionnel)
  - Defines a specific unit measure that can be different from the SRS definition

SQL EXAMPLE

```
select  a.name
from buildings b, streets a
where b.id= 2198376 and
sdo_within_distance
( a.geom,b.geom,'distance=1 UNIT=kilometer')='TRUE';
```

SDO_NN (nearest neighbor)

SDO_NN(<geometry-1>,<geometry-2>,'[paramètres optionnels]') = 'TRUE';

- SDO_NUM_RES (optional)
  - number of neighbors to return (default = 1)
- SDO_BATCH_SIZE (optional)
  - To be used instead of SDO_NUM_RES: it returns subsets of nearest neighbors until all the criteria defined by the « where» condition are satisfied.
- UNIT (optionnel avec l'utilisation de SDO_NN_DISTANCE)
  - Measure unit

SQL EXAMPLE

```
select  s.type
from streets s, polls p
where s.id = 1289 and
sdo_nn(s.geom,p.geom,'sdo_num_res=5 unit=meter',1) = 'TRUE';
```

Oracle Spatial FUNCTIONS

Spatial Functions

- Are not requiring spatial indexes
- Can be used on small tables without indexing
- Can be used either in the « SELECT» or « WHERE » clauses.
- The input geometries must have the same reference system

SDO_GEOM.RELATE

SDO_GEOM.RELATE(
<geometry-1>,'<mask>',<geometry-2>,<tolerance>);

- geometry-1 (searched object)
  - SDO_GEOMETRY object
- geometry-2
  - SDO_GEOMETRY object
- mask
  - Spatial Relationships to be tested
- tolerance
  - Tolérance on the data granularity
- REMINDER:
  - use SDO_RELATE in the where clause because SDO_GEOM.RELATE doesn't use spatial indexing: it is slower

SQL EXAMPLE

```
select c.name, sdo_geom.relate(d.geom,'determine',t.geom,0.5) relation
from districts d, towns t
where d.distname = 'NY' and d.distname = t.district;
```

Oracle Spatial Analysis

Spatial Analysis

- Oracle spatial functionalities are used to analyze spatial objects
- The following examples are using spatial functions and operators in order to compute some basic analysis

Calculate a Surface

SDO_GEOM.SDO_AREA (<geometry>,<tolerance> [, <unit>])

- geometry
  - SDO_GEOMETRY that defines a polygon
- unit
  - Unit of the results: for a list of possible units look at:

```
select sdo_unit from
mdsys.sdo_dist_units;
```

```
select sum(sdo_geom.sdo_area(c1.geom,0.5,'unit=sq_meter')) surface
from towns t1, towns t2
where t2.name = 'New York' and
sdo_relate (t1.geom,t2.geom,
'mask=TOUCH querytype=WINDOW') = 'TRUE';
```

Calculate the Length

SDO_GEOM.SDO_LENGTH (<geometry>,<tolerance> [, <unit>])

```
select name,sdo_geom.sdo_length(geom,0.5,'unit=kilometer') length
from streets
where number = 1234;
```

Calculate the Distance

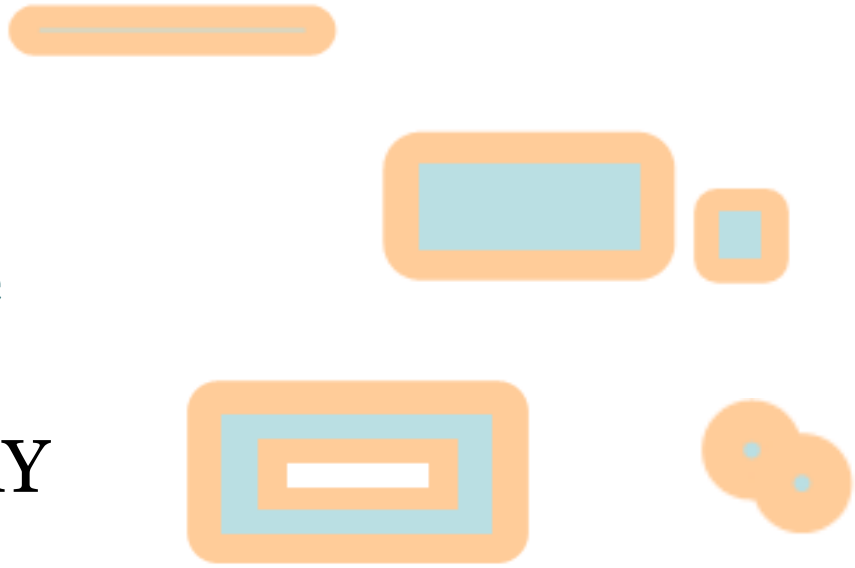
SDO_GEOM.SDO_DISTANCE (<geometry-1>, <geometry-2>,<tolerance> [, <unit>])

```
select sdo_geom.sdo_distance(f1.geom,f2.geom,0.5) distance
from forests f1, forests f2
where f1.id = 1 and f2.id = 2;
```

Buffer Calculation

SDO_GEOM.SDO_BUFFER (<geometry>,<distance>, <tolerance> [, '<params>'])

- geometry
  - SDO_GEOMETRY
- distance
  - Distance of the buffer zone
- Returns: SDO_GEOMETRY

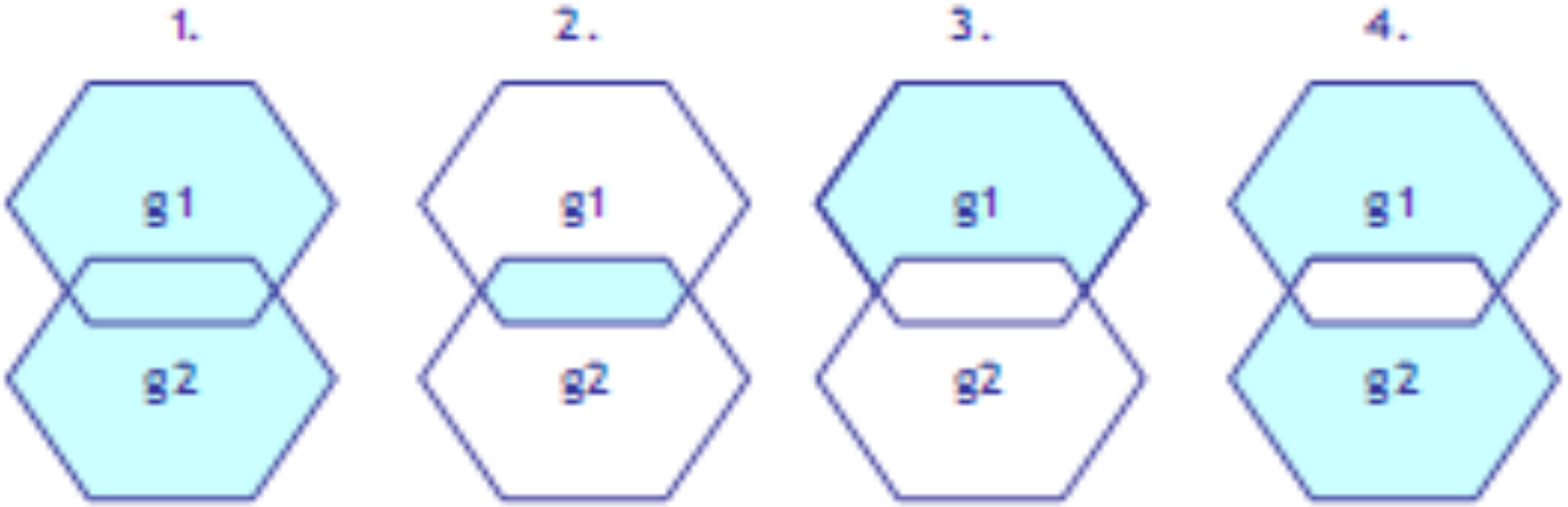


```
select sdo_geom.sdo_buffer (geom,25,0.5,'unit=km')  buffer
from streets
where number = 120;
```

Logic Functions

SDO_GEOM.SDO_XXX (<geometry-1>,<geometry-2>,<tolerance>)

1. SDO_GEOM.SDO_UNION
2. SDO_GEOM.SDO_INTERSECTION
3. SDO_GEOM.SDO_DIFFERENCE
4. SDO_GEOM.SDO_XOR



References

- http://en.wikipedia.org/wiki/Oracle_Spatial
- http://fdo.osgeo.org/
- http://download.oracle.com/docs/cd/B28359_01/appdev.111/b28400/sdo_intro.htm
- Marc Riedo GIS Technology I - EPFL cours slides
- Slides adapted from Simone Campora (simone.campora@epfl.ch) - EFPL