

# SIG. Tema 4: Almacenamiento de *Información Geográfica*

José Samos Jiménez

2020 jsamos (Isi-ugr)  
Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Granada

Curso 2020-21

## Ejemplo: Datos inmobiliarios

Town	Postal code	Street	Number
London	N7 6PA	Holloway Road	32
West Berkshire	RG12 1DF	Charles Square	45
Bristol	BS1 4UZ	St Augustine's Parade	39

Town	Postal code	Street	Number	Latitude	Longitude
London	N7 6PA	Holloway Road	32	51.556173	-0.116190
West Berkshire	RG12 1DF	Charles Square	45	51.381320	-1.344165
Bristol	BS1 4UZ	St Augustine's Parade	39	51.453462	-2.598348

Town	Postal code	Street	Number	the_geom
London	N7 6PA	Holloway Road	32	POINT( -0.116190, 51.556173)
West Berkshire	RG12 1DF	Charles Square	45	POINT( -1.344165, 51.381320)
Bristol	BS1 4UZ	St Augustine's Parade	39	POINT(-2.598348, 51.453462)

- Soporta consultas espaciales.

# Contenido

- 1 Uso de Bases de Datos Espaciales
- 2 Diseño de esquemas
- 3 Representación Geográfica y Nivel Lógico
- 4 Nivel Físico
- 5 Conclusiones
- 6 Bibliografía

## Uso de Bases de Datos Espaciales

- 1 Uso de Bases de Datos Espaciales
- 2 Diseño de esquemas
- 3 Representación Geográfica y Nivel Lógico
- 4 Nivel Físico
- 5 Conclusiones
- 6 Bibliografía

# SGBD Espaciales

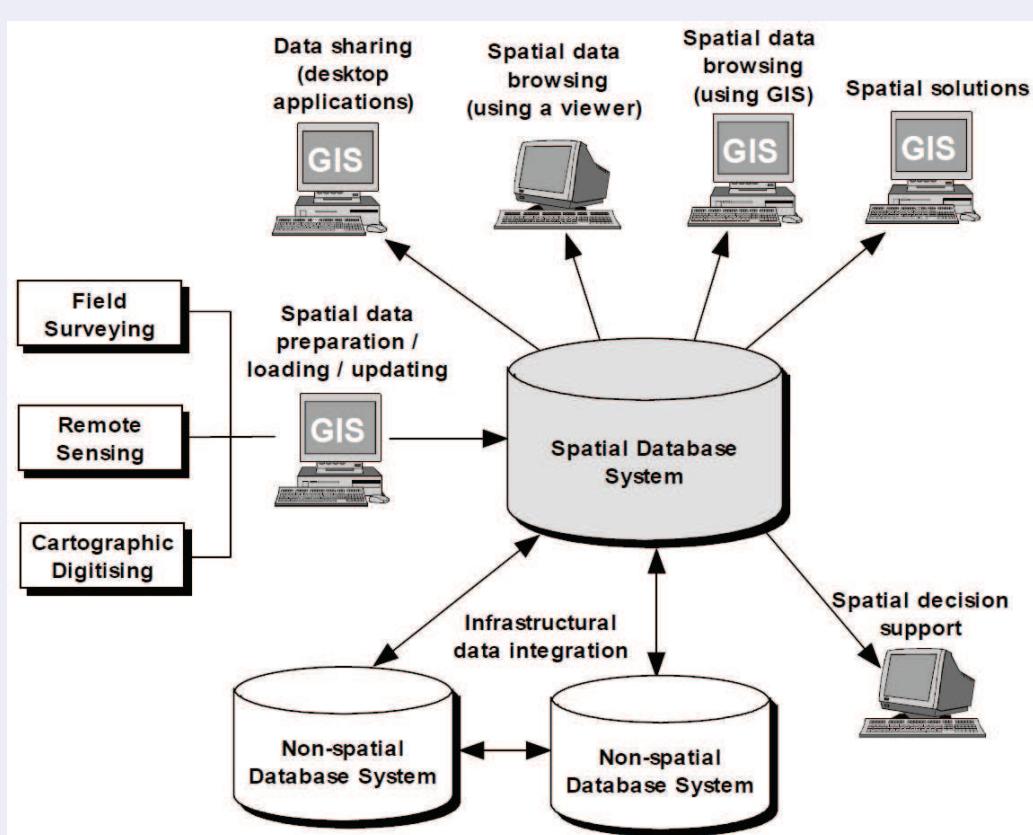
## ¿Por qué usar BD Espaciales?

- Tenemos datos espaciales en archivos, ¿para qué usar una BD?
  - ▶ Por las mismas razones por las que se usan para datos no espaciales.
- Velocidad.
- Soporte multiusuario.
- Soporte a las consultas.
- Facilidad de gestión (redundancia, coste, mantenimiento, seguridad).

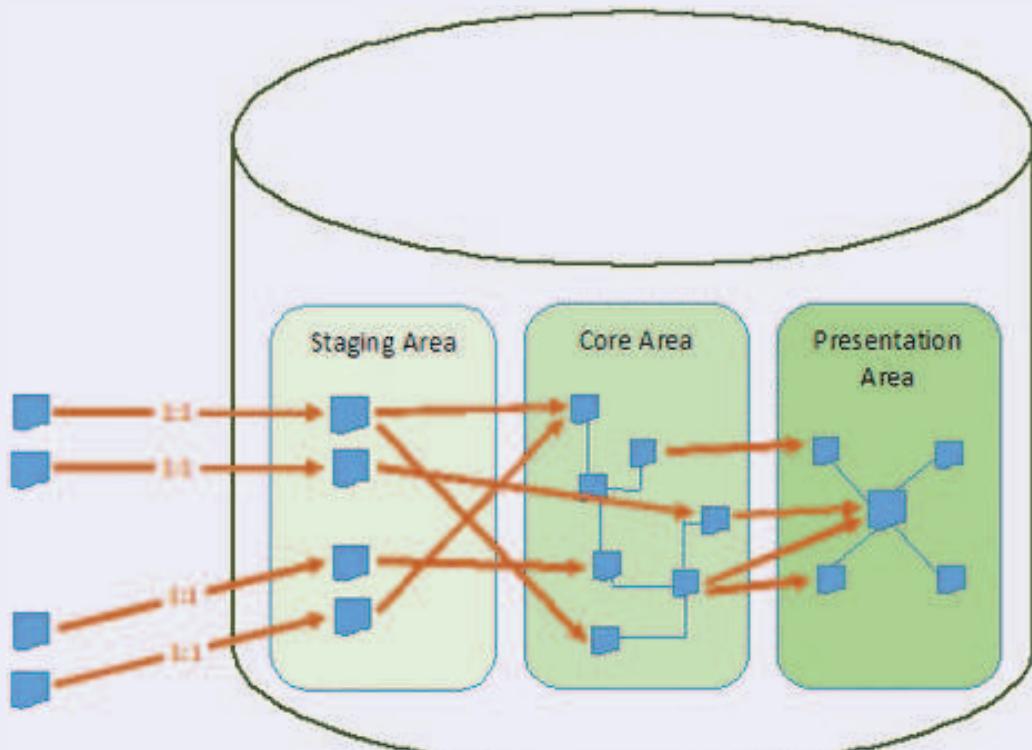
## Ejemplos de SGBD Espaciales

- Oracle Spatial, IBM DB2, Microsoft SQL Server, ArcGIS.
- MySQL, SQLite: SpatiaLite y RasterLite, PostgreSQL: PostGIS.

# SGBD Espaciales y SIG



# Esquemas en la BD



## Diseño de esquemas

1 Uso de Bases de Datos Espaciales

2 Diseño de esquemas

- Niveles de abstracción y modelos de datos
- Proceso de diseño

3 Representación Geográfica y Nivel Lógico

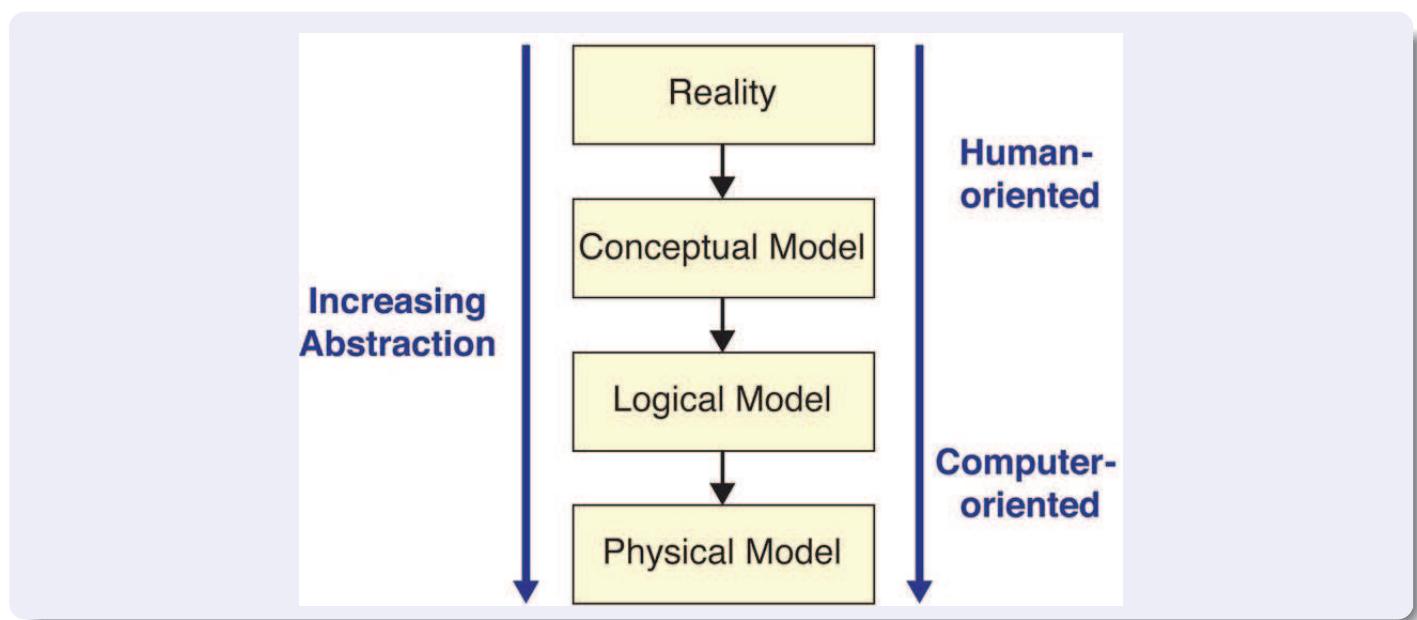
4 Nivel Físico

5 Conclusiones

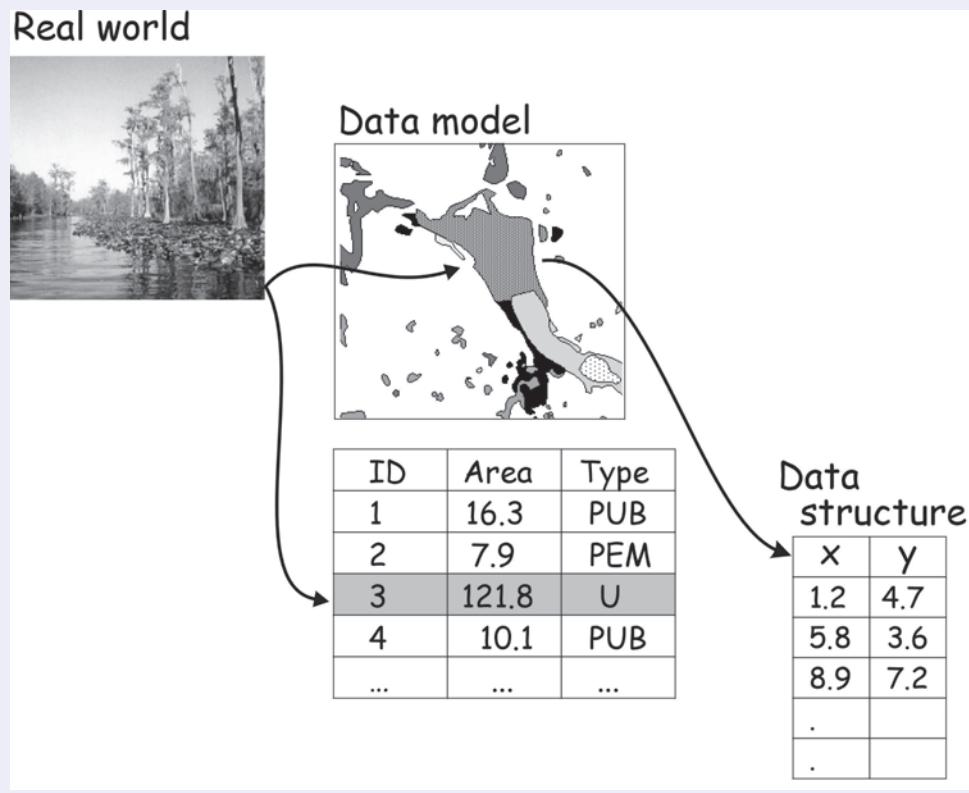
6 Bibliografía

# Niveles de abstracción y modelos de datos

## Niveles de abstracción y modelos de datos

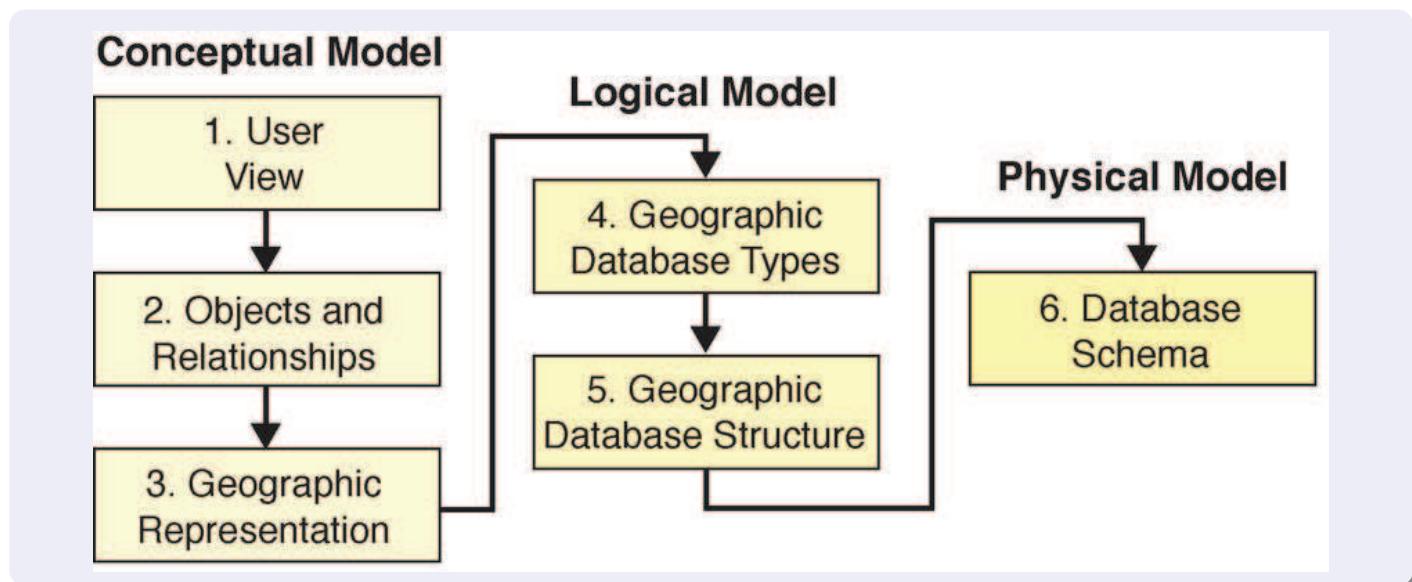


# Modelos de datos geográficos

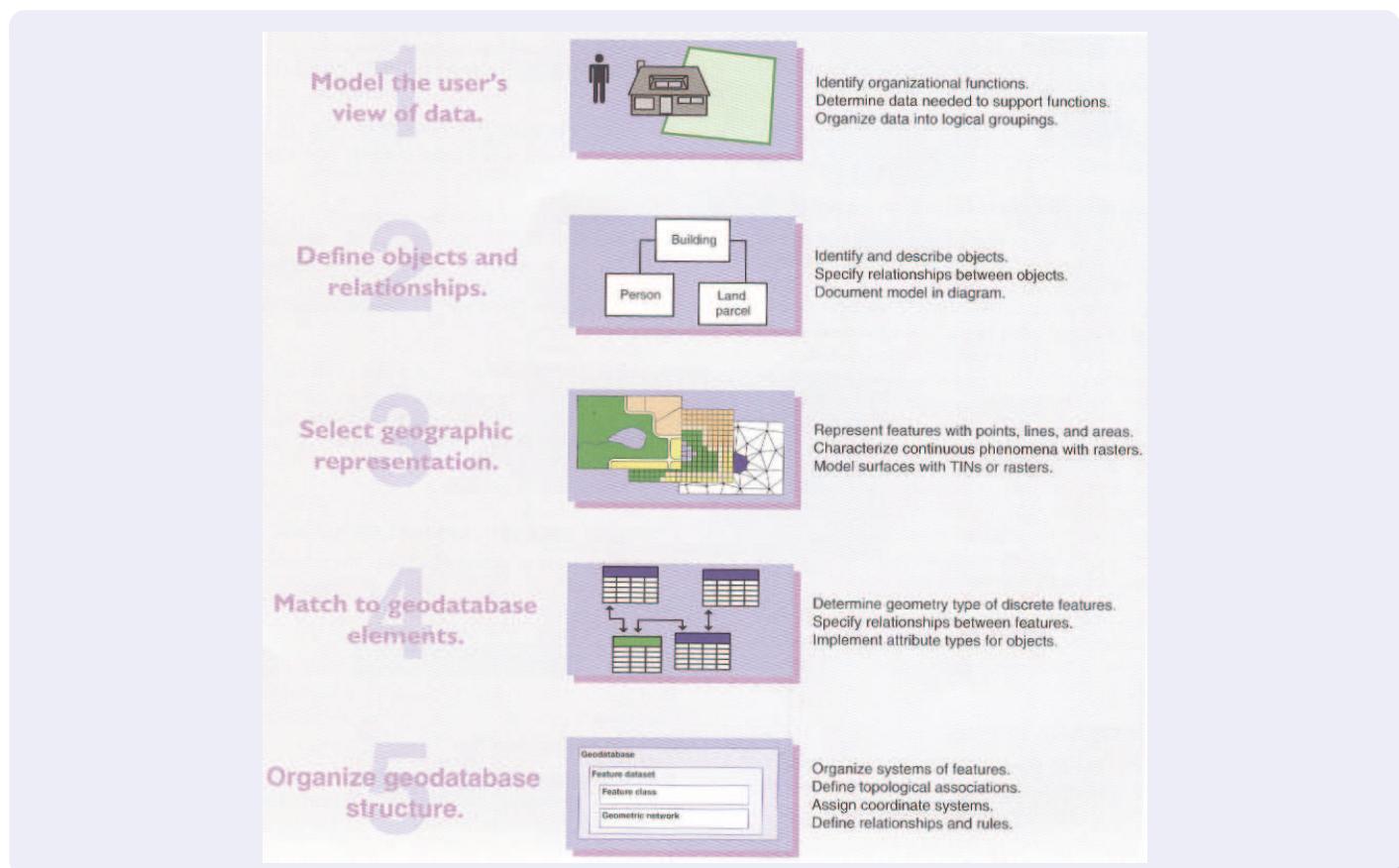


## Proceso de diseño

# Proceso de diseño (i)



# Proceso de diseño (y ii)



# Representación Geográfica y Nivel Lógico

1 Uso de Bases de Datos Espaciales

2 Diseño de esquemas

3 Representación Geográfica y Nivel Lógico

- Ráster vs. Vectorial
- Geometría
- Geografía
- Ráster
- Topología

4 Nivel Físico

5 Conclusiones

6 Bibliografía

## Ráster vs. Vectorial

# Ráster vs. Vectorial (i)

## Planteamiento

- Expresar *qué* y *cómo*: Ráster.
- Expresar *dónde*: Vectorial.

## Tipo de variable

- Continua: Ráster.
- Discreta: Vectorial.

## Tipo de vista del espacio geográfico

- Vista de objeto: Vectorial.
  - ▶ Localizado exactamente, acotado, contable.
- Vista de campo: Ráster.
  - ▶ Superficie continua con atributos definidos en cada posición.

# Ráster vs. Vectorial (y ii)

Characteristic	Raster	Vector
data structure	usually simple	usually complex
storage requirements	larger for most data sets without compression	smaller for most data sets
coordinate conversion	may be slow due to data volumes, and require resampling	simple
analysis	easy for continuous data, simple for many layer combinations	preferred for network analyses, many other spatial operations more complex
spatial precision	floor set by cell size	limited only by positional measurements
accessibility	easy to modify or program, due to simple data structure	often complex
display and output	good for images, but discrete features may show "stairstep" edges	maplike, with continuous curves, poor for images

# Modelo de datos Vectorial (del Tema 1)

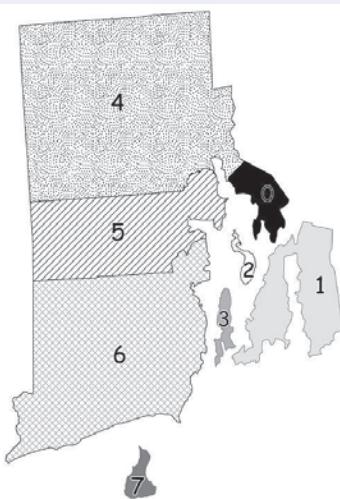
Es una abstracción de la realidad (interpretación o generalización).

- *Punto*: representado por su posición (coordenadas).
  - ▶ Para modelizar ocurrencias o eventos: ciudades, restaurantes, aeropuertos, casos de una enfermedad.
  - ▶ Análisis: análisis de patrones de puntos (interrelación de eventos).
- *Línea*: definida por una secuencia de puntos.
  - ▶ Para representar fenómenos estáticos: carreteras, ríos.
  - ▶ Para representar datos dinámicos: rutas por carretera, por aire, distancias.
  - ▶ Análisis: relación con puntos y áreas.
- *Polígono/área*: forma cerrada definida mediante líneas.
  - ▶ Para representar bordes: continentes, países, regiones, campos.
  - ▶ Análisis: relación con puntos y líneas.

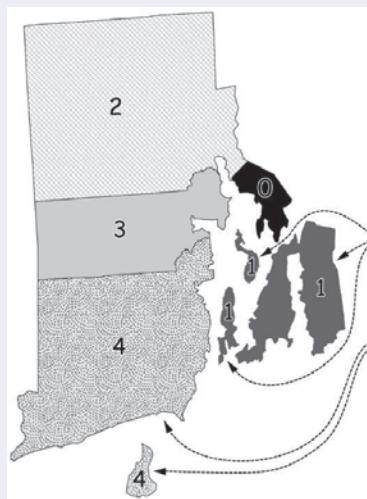
Un área se sumariza en un punto mediante su *centroide*.

## Vectorial: relación con la tabla de atributos (i)

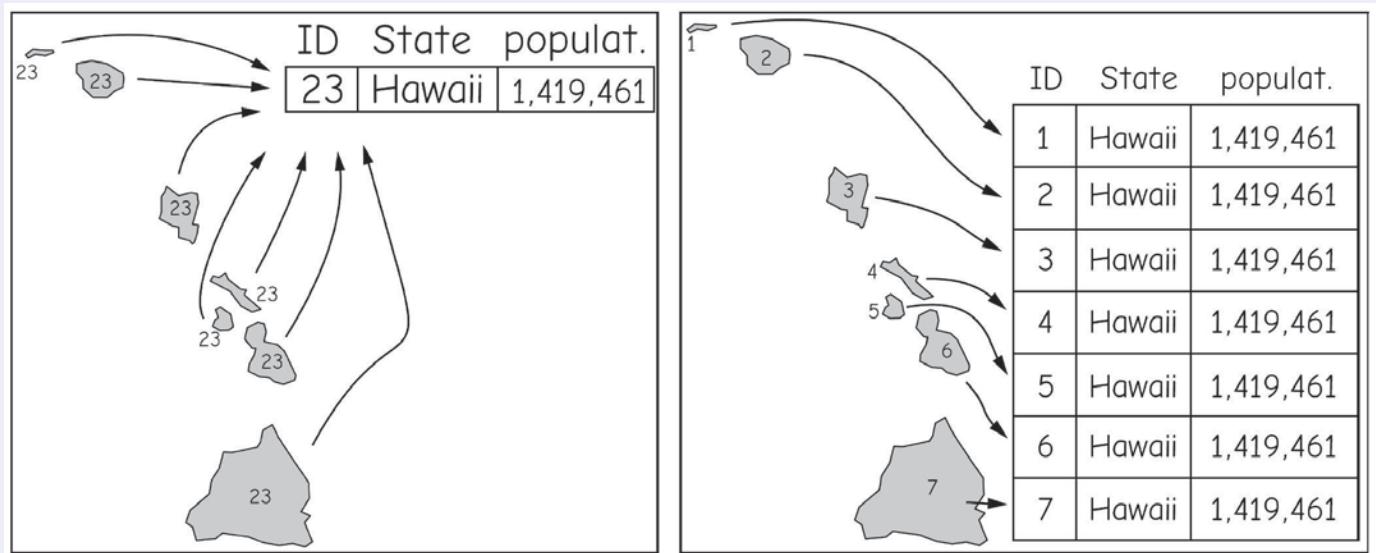
### Uno a uno



### Muchos a uno



# Vectorial: relación con la tabla de atributos (y ii)



¡Cuidado!

## Ráster: relación con la tabla de atributos

a) Raster, one-to-one

A	A	A	A	B	B	B	B	B	B
A	A	A	A	B	B	B	B	B	B
A	A	A	A	B	B	B	B	B	B
A	A	A	B	B	B	B	B	B	B
A	A	A	C	C	B	B	B	B	B
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	D	D
C	C	C	C	C	D	D	D	E	E
C	C	C	C	C	D	D	E	E	E

attribute table  
(cell 1 is upper-left corner)

cell-ID	IDorg	class	area
1	A	10	0.8
2	A	10	0.8
3	A	10	0.8
4	A	10	0.8
5	B	11	0.8
6	B	11	0.8
7	B	11	0.8
.	.	.	.
.	.	.	.
100	E	10	0.8

b) Raster, many-to-one

10	10	10	10	10	11	11	11	11	11
10	10	10	10	10	11	11	11	11	11
10	10	10	10	10	11	11	11	11	11
10	10	10	10	11	11	11	11	11	11
10	10	10	10	15	15	11	11	11	11
15	15	15	15	15	15	21	21	21	21
15	15	15	15	15	15	21	21	21	21
15	15	15	15	15	15	21	21	21	21
15	15	15	15	15	15	21	21	21	10
15	15	15	15	15	15	21	21	10	10

attribute table

class	area
10	18.4
11	24.0
15	21.6
21	13.6

# Tipos de datos espaciales

## Geometría

- Usa el plano cartesiano.
- Estándar *ISO/IEC 13249-3 SQL/MM Part 3: Spatial*

## Geografía

- Usa la superficie curva del modelo de la Tierra.
- No es estándar. Lo usan *PostGIS* y *SQL Server*.

## Ráster

- Soportado por la mayoría de SGBD Espaciales.

## Topología

- Modela el mundo como una red de nodos, bordes y caras.
- Adecuado para:
  - ▶ parcelas, carreteras, divisiones administrativas, jurisdicciones, etc.

## Geometría

# Geometría

```
CREATE TABLE my_linestrings (
    id serial PRIMARY KEY,
    name varchar(20),
    my_linestrings geometry(LINESTRING)
);
```

## Elementos básicos

- Punto.
- Línea.
- Polígono.

Con Z, M, ZM.

## Características

- Considera el mundo como un plano cartesiano: las líneas paralelas...
- Basado en Geometría Analítica.
- Buena velocidad computacional.

## Ejemplo: *Puntos*

```
CREATE TABLE my_points (
    id serial PRIMARY KEY,
    p geometry(POINT),
    pz geometry(POINTZ),
    pm geometry(POINTM),
    pzm geometry(POINTZM),
    p_srid geometry(POINT, 4269)
);
INSERT INTO my_points (p, pz, pm, pzm, p_srid)
VALUES (
    ST_GeomFromText('POINT(1 -1)'),
    ST_GeomFromText('POINT Z(1 -1 1)'),
    ST_GeomFromText('POINT M(1 -1 1)'),
    ST_GeomFromText('POINT ZM(1 -1 1 1)'),
    ST_GeomFromText('POINT(1 -1)', 4269)
);
```

- *SRID*: Tabla *spatial\_ref\_sys*.

# Definición de estándares

## OGC (*Open Geospatial Consortium*)

- Definición de tipos de datos geográficos y funciones para SGBD.
- OGC/SFSQL (*Spatial Features for SQL*): Acceso mediante SQL.

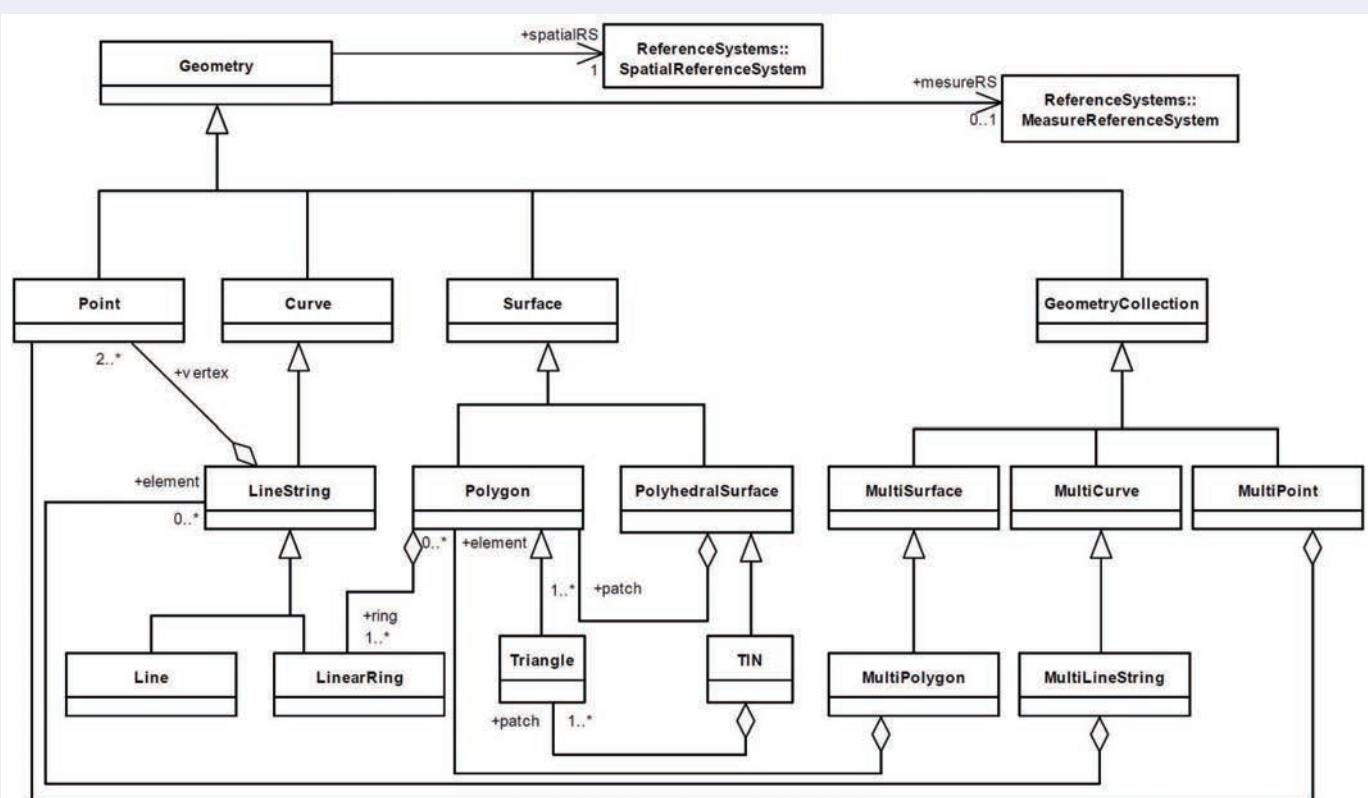
## OSGeo (*Open Source Geospatial Fundation*)

- Participa en la definición de estándares (con OGC e ISO).
- Algunos proyectos:
  - ▶ QGIS, PostGIS, GDAL/OGR, GeoServer, OpenLayers.

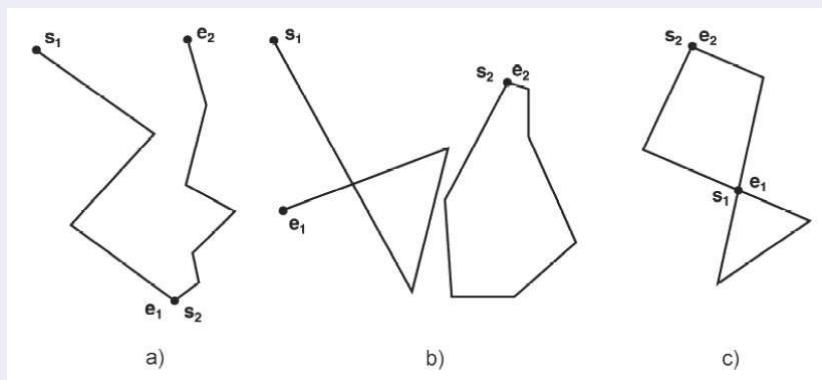
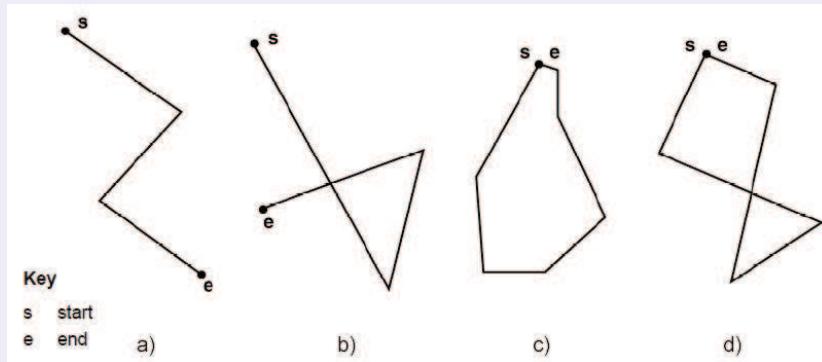
## ANSI/ISO SQL y *SQL/MM Part 3: Spatial*

- Jerarquía de clases.
- Funciones estándar para datos espaciales en SQL.

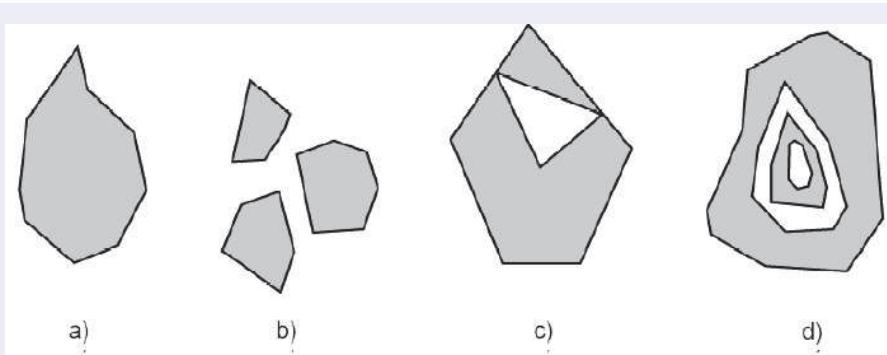
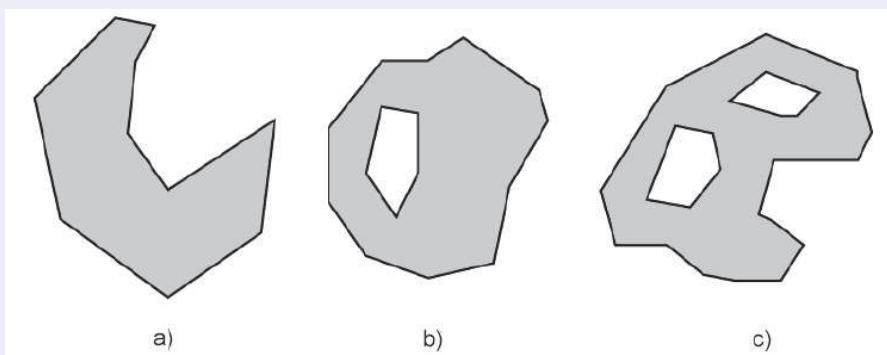
## Jerarquía de clases



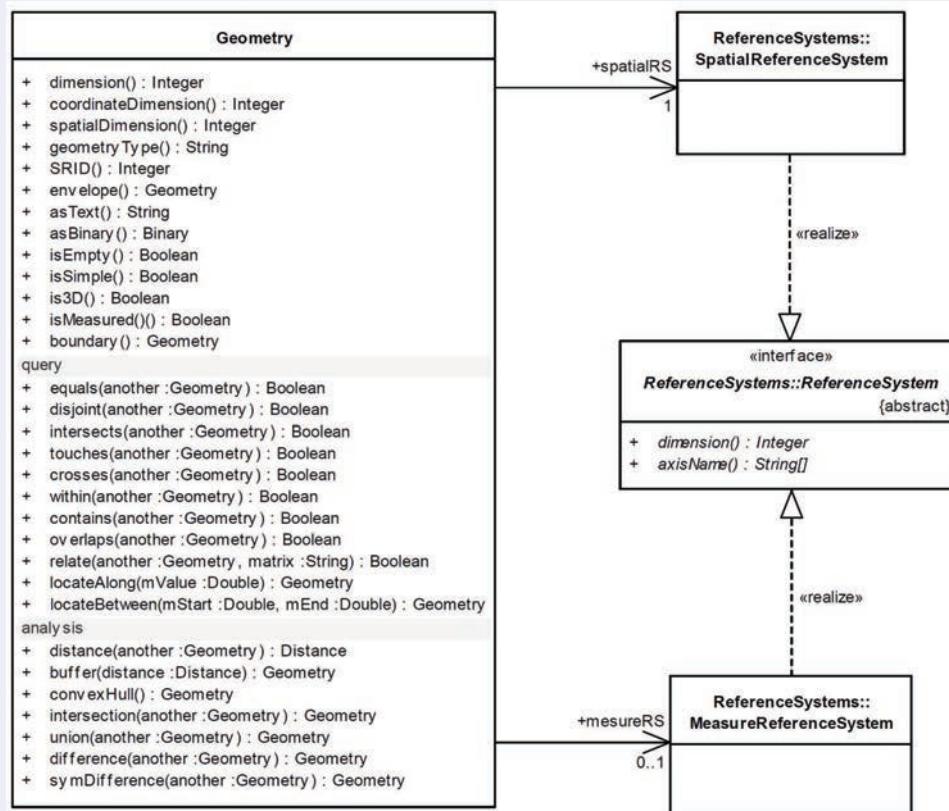
## Ejemplos de *LineString* y *MultiLineString*



## Ejemplos de *Polygon* y *MultiPolygon*



# Operaciones de la clase Geometry



## Operadores Booleanos de relaciones espaciales

**query**

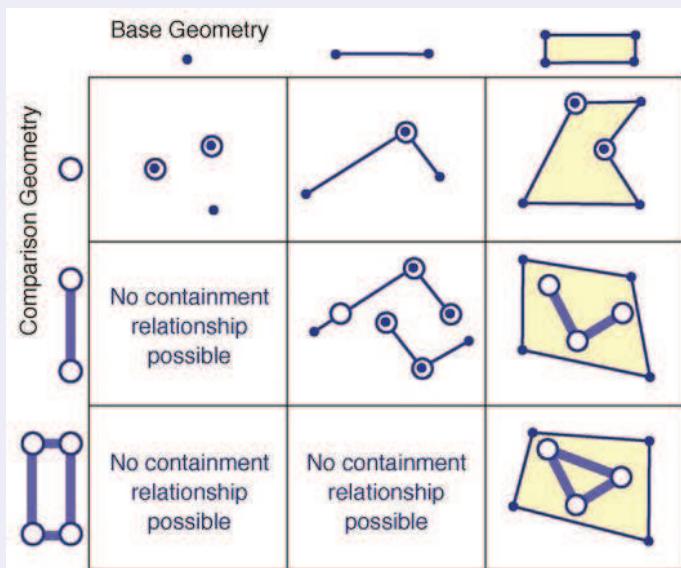
- + equals(another : Geometry) : Boolean
- + disjoint(another : Geometry) : Boolean
- + intersects(another : Geometry) : Boolean
- + touches(another : Geometry) : Boolean
- + crosses(another : Geometry) : Boolean
- + within(another : Geometry) : Boolean
- + contains(another : Geometry) : Boolean
- + overlaps(another : Geometry) : Boolean
- + relate(another : Geometry, matrix : String) : Boolean

**Contains** (anotherGeometry: Geometry): Integer — Returns 1 (TRUE) if *this* geometric object "spatially contains" anotherGeometry.

**Relate** (anotherGeometry: Geometry, intersectionPatternMatrix: String): Integer — Returns 1 (TRUE) if *this* geometric object is spatially related to anotherGeometry by testing for intersections between the interior, boundary and exterior of the two geometric objects as specified by the values in the intersectionPatternMatrix. This returns FALSE if all the tested intersections are empty except exterior (*this*) intersect exterior (another).

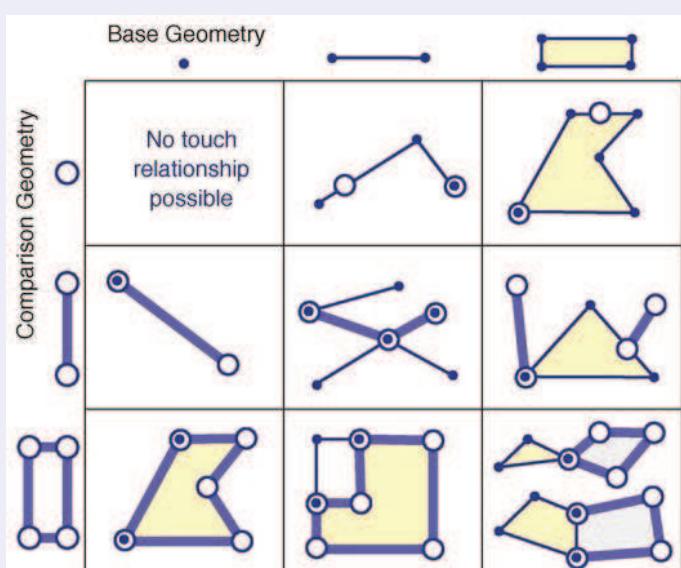
## Ejemplos del operador *Contains*

**Contains** (anotherGeometry: Geometry): Integer — Returns 1 (TRUE) if *this* geometric object "spatially contains" anotherGeometry.



## Ejemplos del operador *Touches*

**Touches** (anotherGeometry: Geometry): Integer — Returns 1 (TRUE) if *this* geometric object "spatially touches" anotherGeometry.



# Métodos de soporte al análisis espacial

## analysis

- + distance(another :Geometry) : Distance
- + buffer(distance :Distance) : Geometry
- + convexHull() : Geometry
- + intersection(another :Geometry) : Geometry
- + union(another :Geometry) : Geometry
- + difference(another :Geometry) : Geometry
- + symDifference(another :Geometry) : Geometry

**Distance** (anotherGeometry: Geometry):Double — Returns the shortest distance between any two Points in the two geometric objects as calculated in the spatial reference system of *this* geometric object. Because the geometries are closed, it is possible to find a point on each geometric object involved, such that the distance between these 2 points is the returned distance between their geometric objects.

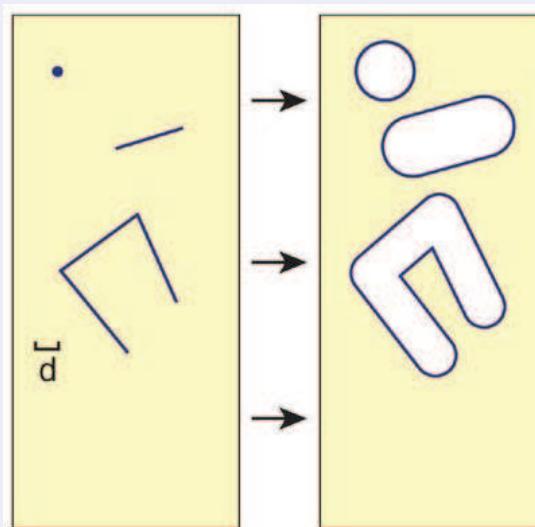
...calculada en el sistema de referencia espacial del objeto geométrico.

**SymDifference** (anotherGeometry: Geometry): Geometry — Returns a geometric object that represents the Point set symmetric difference of *this* geometric object with anotherGeometry.

Está en una o en otra, no en las dos.

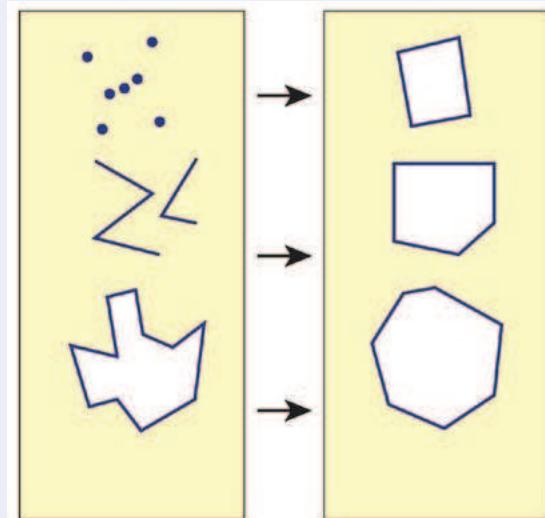
## Ejemplos del método Buffer

**Buffer** (distance: Double): Geometry — Returns a geometric object that represents all Points whose distance from *this* geometric object is less than or equal to distance. Calculations are in the spatial reference system of *this* geometric object. Because of the limitations of linear interpolation, there will often be some relatively small error in this distance, but it should be near the resolution of the coordinates used.



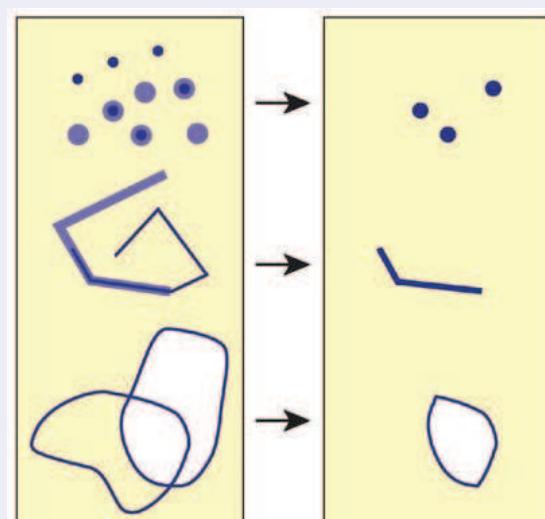
## Ejemplos del método *ConvexHull*

**ConvexHull ( )**: Geometry — Returns a geometric object that represents the convex hull of *this* geometric object. Convex hulls, being dependent on straight lines, can be accurately represented in linear interpolations for any geometry restricted to linear interpolations.



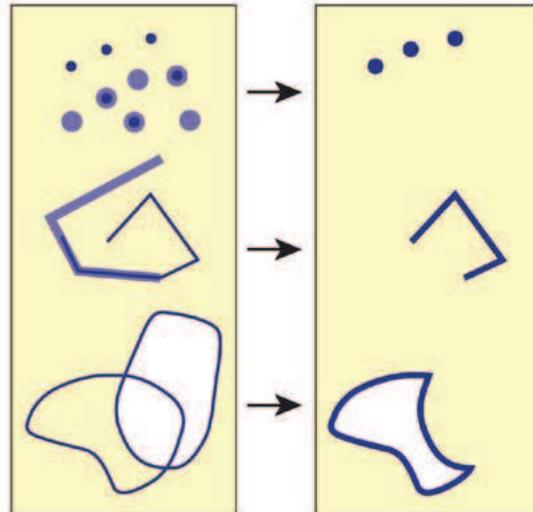
## Ejemplos del método *Intersection*

**Intersection** (*anotherGeometry*: Geometry): Geometry — Returns a geometric object that represents the Point set intersection of *this* geometric object with *anotherGeometry*.



# Ejemplos del método *Difference*

**Difference (anotherGeometry: Geometry): Geometry** — Returns a geometric object that represents the Point set difference of *this* geometric object with anotherGeometry.

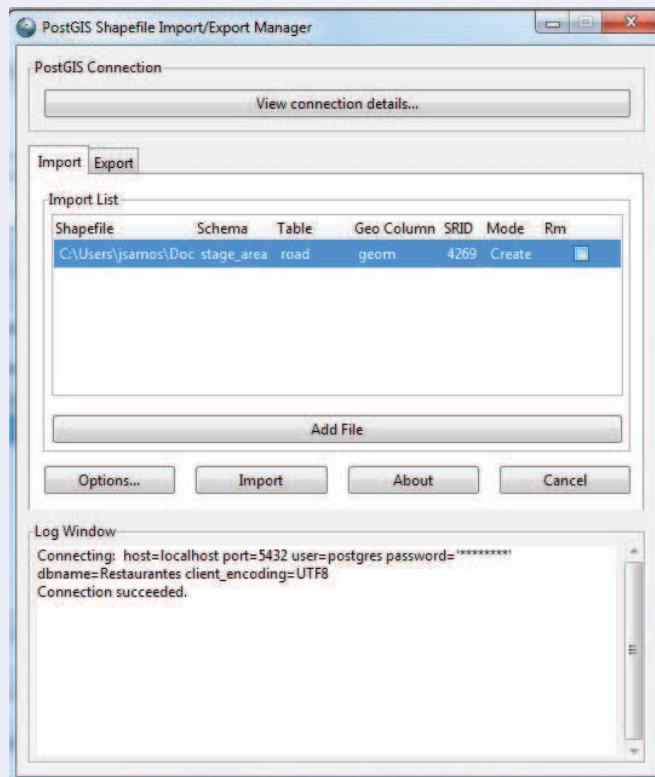


## Clasificación de funciones [OH15]

- *Output functions*—These functions output spatial object representations in various standard formats (WKT, WKB, GML, SVG, KML, GeoJSON).
- *Constructor functions*—These functions create PostGIS spatial objects from a well-known format, such as a well-known text (WKT) format, a well-known binary (WKB) format, or Geography JavaScript Object Notation (GeoJSON).
- *Accessor and setter functions*—These functions work against a single spatial object and return or set attributes of the object.
- *Measurement functions*—These functions return scalar measurements of a spatial object.
- *Decomposition functions*—These functions extract other spatial objects from an input spatial object.
- *Composition functions*—These functions stitch, splice, or group together spatial objects.
- *Simplification functions*—Sometimes you don't need the full resolution of a spatial object. These functions, in the case of geometry and geography, return simplified representations by removing points or linestrings or by rounding the coordinates. The resultant spatial objects still have the basic look and feel of the originals but contain fewer points or elements of lower precision.

# Ejemplo: Cargar una capa vectorial en una nueva tabla

## Interfaz gráfica para *shp2pgsql*



Geografía

# Geografía

```
CREATE TABLE my_point (
    id serial PRIMARY KEY,
    name varchar(20),
    my_point geography(POINT)
);
```

- Usa la superficie curva del modelo de la Tierra.
  - ▶ WGS 84 lon/lat SRID 4326 (sistema de referencia por defecto).
  - ▶ Soporta cualquier sistema de referencia basado en longitud y latitud.
  - ▶ Cálculo de distancias en una esfera.
- Adecuado para representar navegación marítima o aérea, y continentes.
- Tipos de datos y funciones similares a los del tipo Geometría pero basados en coordenadas geodéticas.
  - ▶ Geog en lugar de Geom.
- No es estándar.
  - ▶ Lo usan *PostGIS* y *SQL Server*. En ambos son muy similares.

## Geografía vs. Geometría

### Grados / metros

```
SELECT ST_Distance(
    ST_GeometryFromText('POINT(-118.4079 33.9434)', 4326), -- Los Angeles (LAX)
    ST_GeometryFromText('POINT(2.5559 49.0083)', 4326)      -- Paris (CDG)
);
```

121.898285970107

```
SELECT ST_Distance(
    ST_GeographyFromText('POINT(-118.4079 33.9434)'), -- Los Angeles (LAX)
    ST_GeographyFromText('POINT(2.5559 49.0083)')        -- Paris (CDG)
);
```

9124665.26917268

```
SELECT ST_Distance(ST_Point(0,180)::geography, ST_Point(0,-180)::geography)
```

```
SELECT ST_Distance(ST_Point(0,180)::geometry, ST_Point(0,-180)::geometry)
```

# Transformaciones

## Convertir una columna *Geometry* en *Geography*

```
ALTER TABLE osm_roads
ALTER COLUMN way TYPE geography(MULTIPOLYGON, 4326)
USING ST_Transform(way, 4326) ::geography;
```

- Transformación + conversión.

## Cambiar el *SRID* de una columna

```
ALTER TABLE us_states
ALTER COLUMN geom TYPE geometry(MULTIPOLYGON, 4326)
USING ST_SetSRID(geom, 4326);
```

# Ráster

```
CREATE TABLE my_rasters (
    rid SERIAL PRIMARY KEY,
    name varchar(150),
    rast raster
);
```

- Columna de tipo *raster*.
  - ▶ Se descompone en forma de “azulejos” (*tiles*), del mismo tamaño.
- Pueden tener hasta 255 bandas.
- Son más que fotografías.
- Pueden estar georreferenciados, se les puede asignar un *SRID*.
- Los pixels han de ser números, el mismo tipo en una banda:
  - 1-bit Boolean, abbreviated as 1BB
  - Unsigned integer of 2, 8, 16, or 32 bits, abbreviated as 2BUI, 8BUI, 16BUI, 32BUI
  - Signed integers of 8, 16, or 32 bits, abbreviated as 8BSI, 16BSI, 32BSI
  - Two float types of 32 bits and 64 bits, abbreviated as 32BF and 64BF

## Clasificación de funciones [OH15]

- Constructor functions.
- Output functions.
- Raster band and pixel accessors and setters:
  - ▶ for intrinsic attributes (width, height, scale, etc.).
- Georeferencing functions:
  - ▶ fundamental changes, such as with and height.
- Reclassing functions.
- Polygonizing functions:
  - ▶ conversion of rasters to polygons.

Ejemplo: Cargar un ráster en una nueva tabla

## Comando

```
raster2pgsql GBOOverview.tif -F overview > overview.sql
```

- F: almacenar el nombre del archivo.

## Resultado

# Topología

# Topología

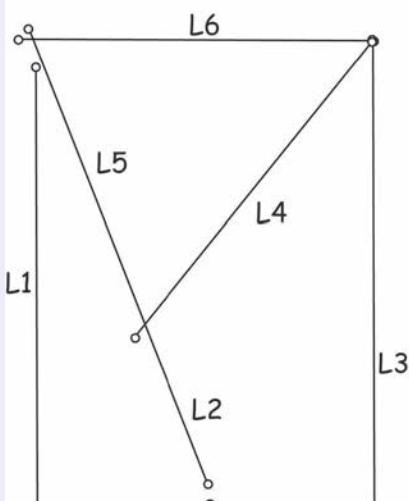
```
CREATE SCHEMA ri;
CREATE TABLE ri.roads(gid serial PRIMARY KEY, road_name text);
SELECT topology.AddTopologyColumn('ri_topo', 'ri', 'roads', 'topo', 'LINE');
```

## Características

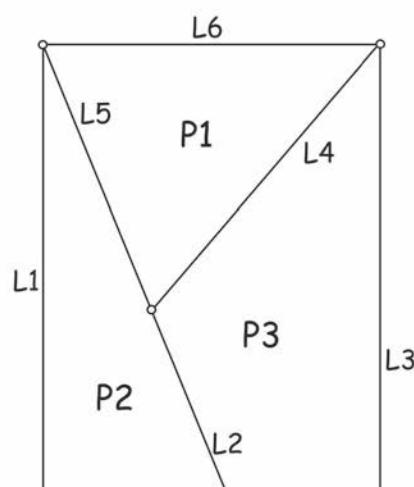
- División exhaustiva y mutuamente exclusiva del espacio.
- Se definen reglas sobre cómo se interrelacionan las geometrías.
- Se consigue simplificación en los objetos modelados.
- Se detectan y previenen problemas.

## Modelo Spaghetti y Modelo Topológico

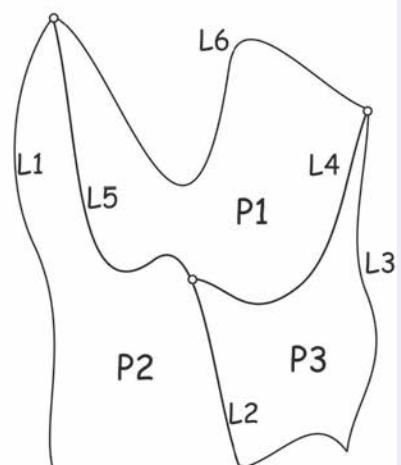
a) spaghetti



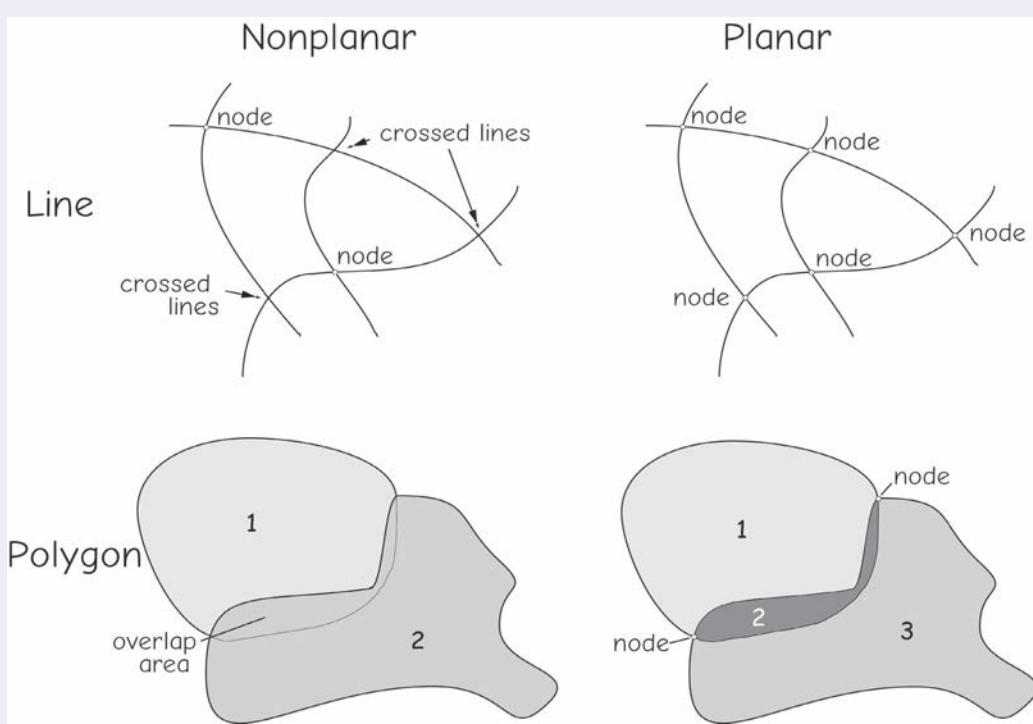
b) topological



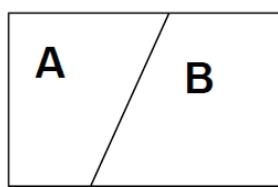
c) topological - warped



# Modelo Topológico *no-plano* y *plano*

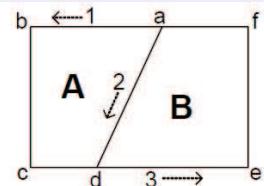


## Modelo no-Topológico vs. Topológico



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

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



Polygon File		Arc File	
Poly_ID	Arcs	Arc_ID	Vertices
A	1, 2	1	b, c
B	2, 3	2	-
		3	e, f

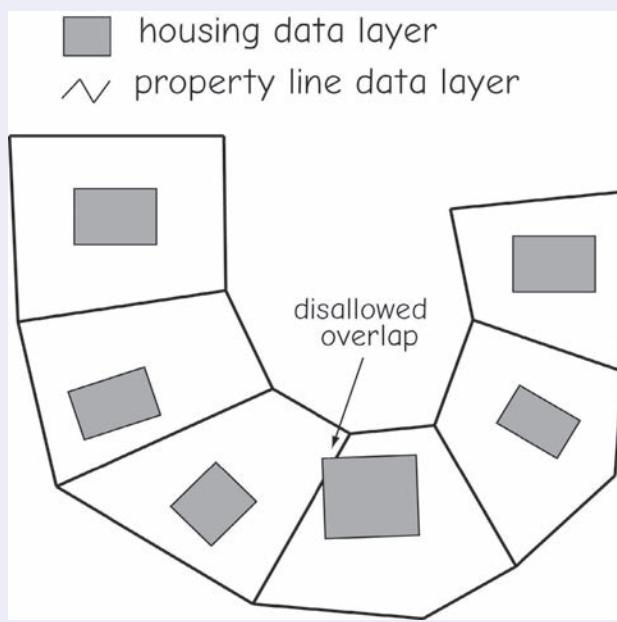
Node File		
Node_ID	X	Y
a	403600	275700
d	403300	275000

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

Network Topology File		Polygon Topology File			
Arc_ID	F_node	T_node	Arc_ID	L_Poly	R_Poly
1	a	d	1	A	World
2	d	a	2	A	B
3	d	a	3	B	World

- Bordes y áreas compartidos se modelan como compartidos.
- Se pueden detectar y prevenir problemas.

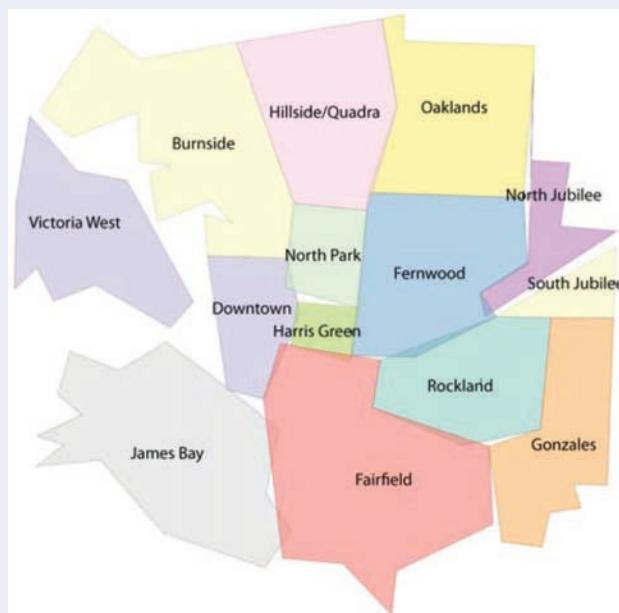
# Reglas del Modelo Topológico entre capas



## Simplificación basada en Geometría

Tolerancia de 150 m.

```
SELECT feat_name, ST_Simplify(topo::geometry,150) As geom_simp  
FROM neighbourhoods;
```



# Simplificación basada en Topología

Tolerancia de 150 m.

```
SELECT feat_name, ST_Simplify(topo,150) As topo_simp  
FROM neighbourhoods;
```



## PostGIS / PostgreSQL *pgRouting*

(<https://pgrouting.org/>)

*pgRouting* extiende PostGIS / PostgreSQL para proporcionar funcionalidad de enrutamiento geoespacial.

### Algoritmos que implementa

- All Pairs Shortest Path, Johnson's Algorithm
- All Pairs Shortest Path, Floyd-Warshall Algorithm
- Shortest Path A\*
- Bi-directional Dijkstra Shortest Path
- Bi-directional A\* Shortest Path
- Shortest Path Dijkstra
- Driving Distance
- K-Shortest Path, Multiple Alternative Paths
- K-Dijkstra, One to Many Shortest Path
- Traveling Sales Person
- Turn Restriction Shortest Path (TRSP)

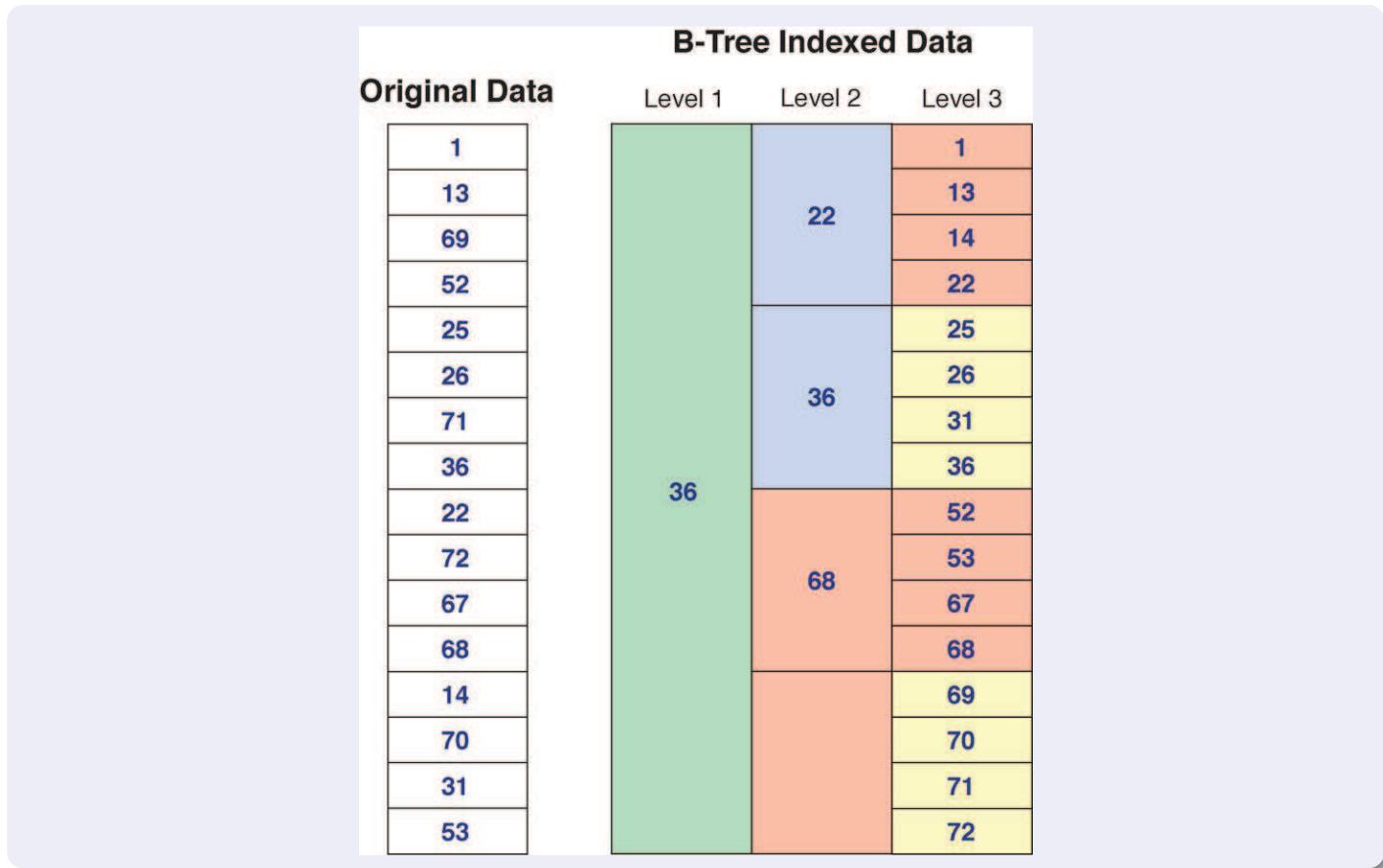


# Nivel Físico

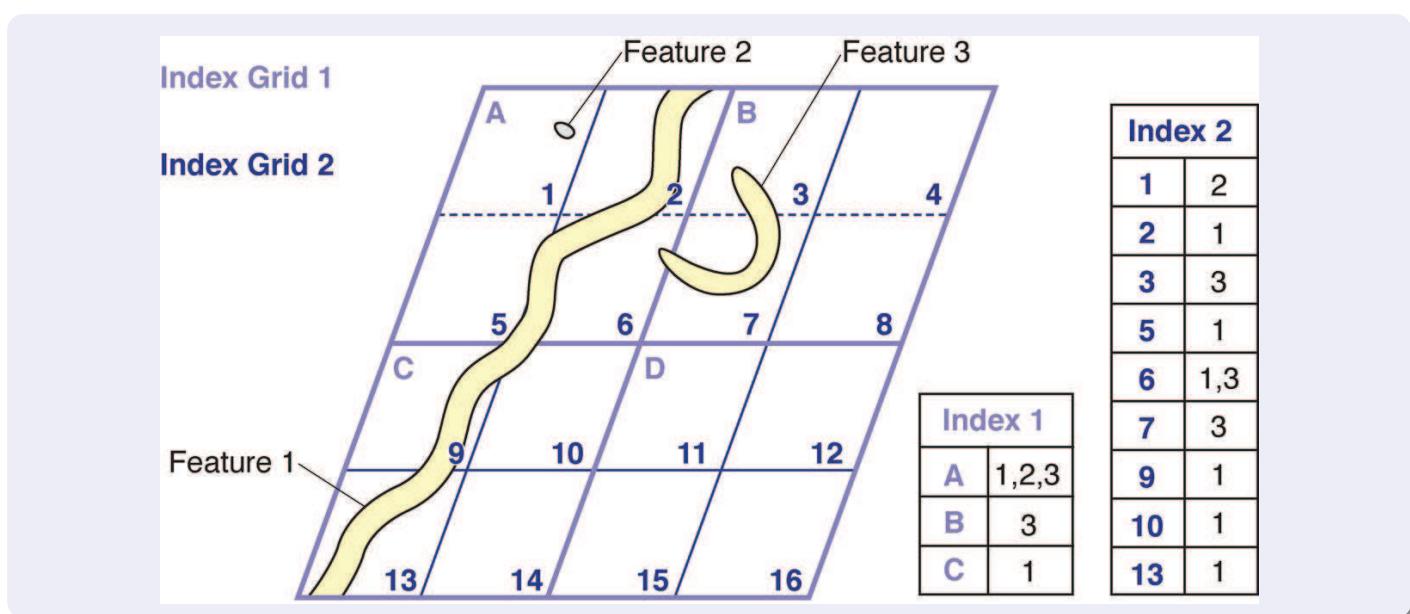
- 1 Uso de Bases de Datos Espaciales
- 2 Diseño de esquemas
- 3 Representación Geográfica y Nivel Lógico
- 4 Nivel Físico
  - Indexación
  - Almacenamiento de Ráster
  - Modelo Topológico
- 5 Conclusiones
- 6 Bibliografía

## Indexación

# Índice *B*-Tree

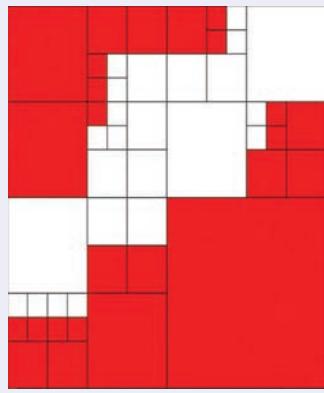
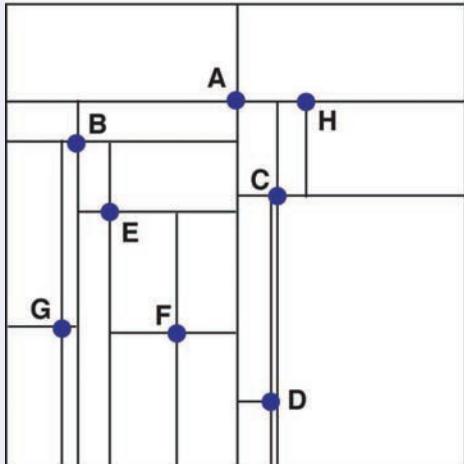
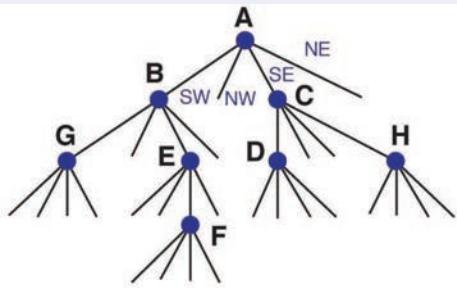


# Índice *Grid* multinivel



- El resultado depende de la relación entre el tamaño de la cuadrícula y el de los objetos y su densidad.

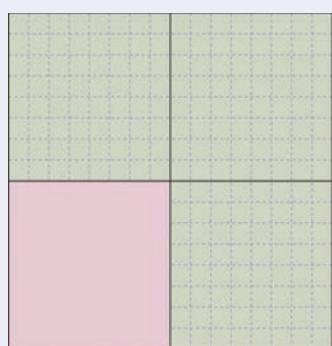
# Índice Quadtree (i)



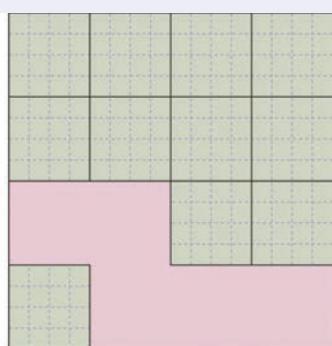
- También para comprimir.
- Tipos: Según el tipo de dato, el algoritmo para dividir y la granularidad.
- Tipos de datos: Puntos, líneas, áreas, superficies, rasters.

# Índice Quadtree (y ii)

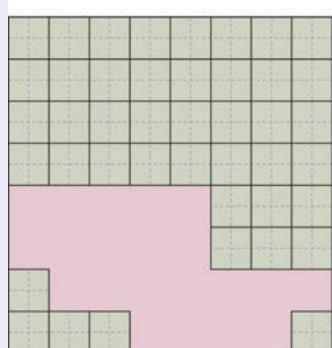
Da soporte a distintos niveles de resolución



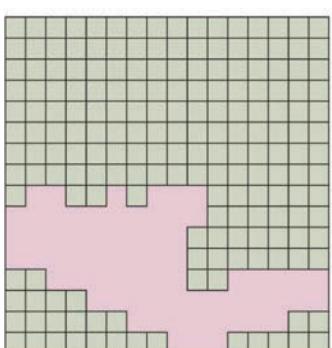
(a)



(b)

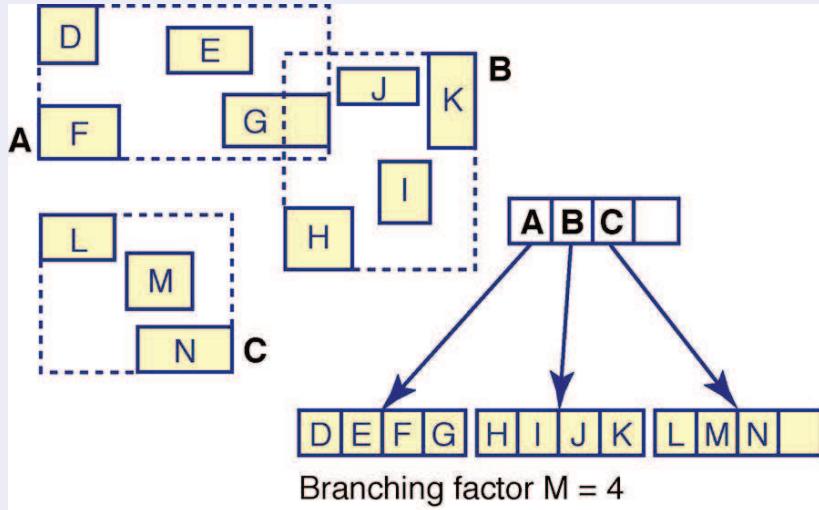
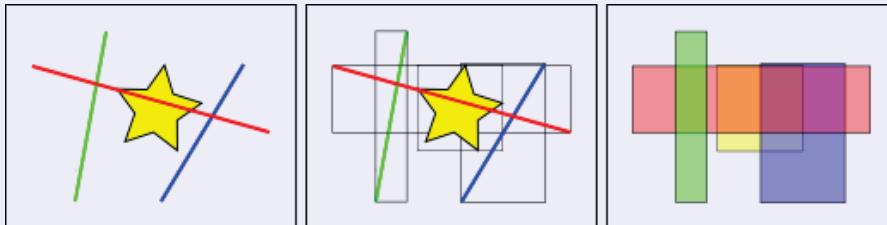


(c)



(d)

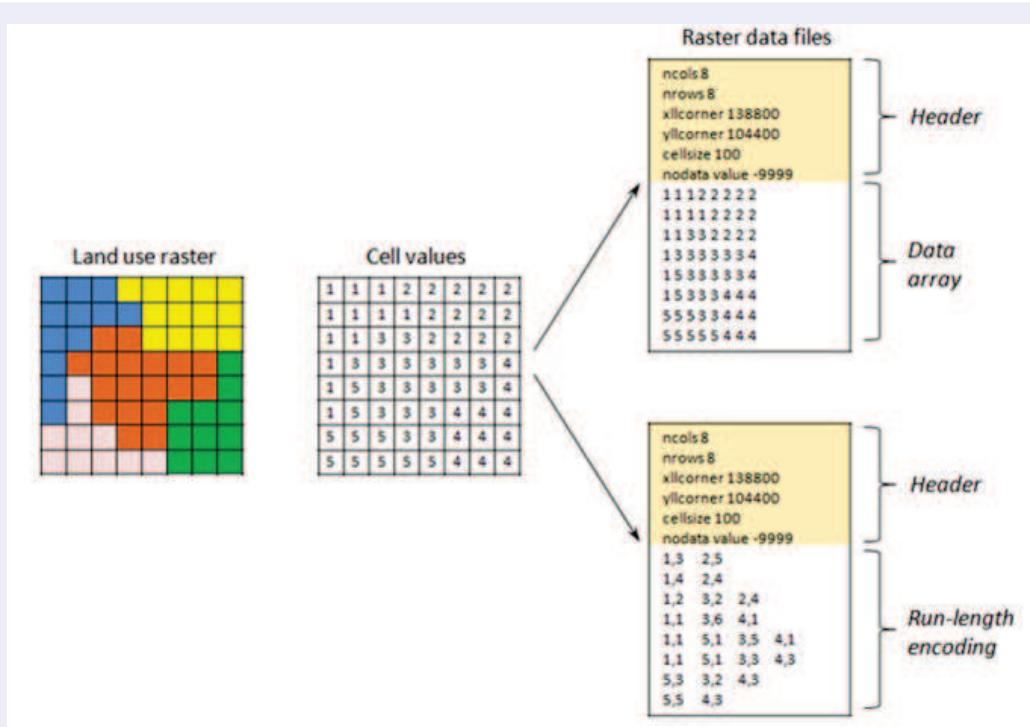
# Índice R-tree y MBR (Minimum Bounding Rectangle)



- Se selecciona el rectángulo que requiera menor expansión.
- Si se expande más allá de un límite, se divide en dos.
- Los usan IBM, Oracle y PostGIS.

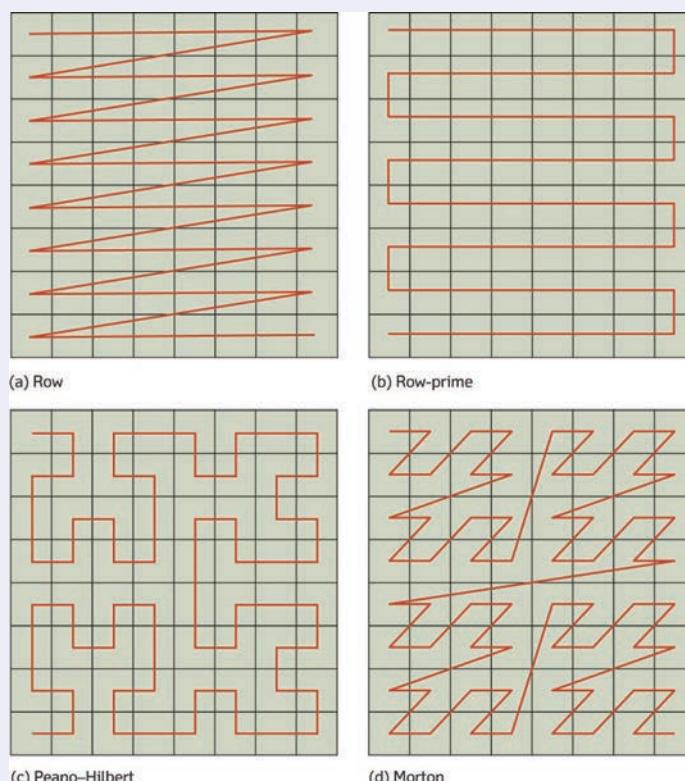
## Almacenamiento de Ráster

# Formas de almacenamiento de celdas



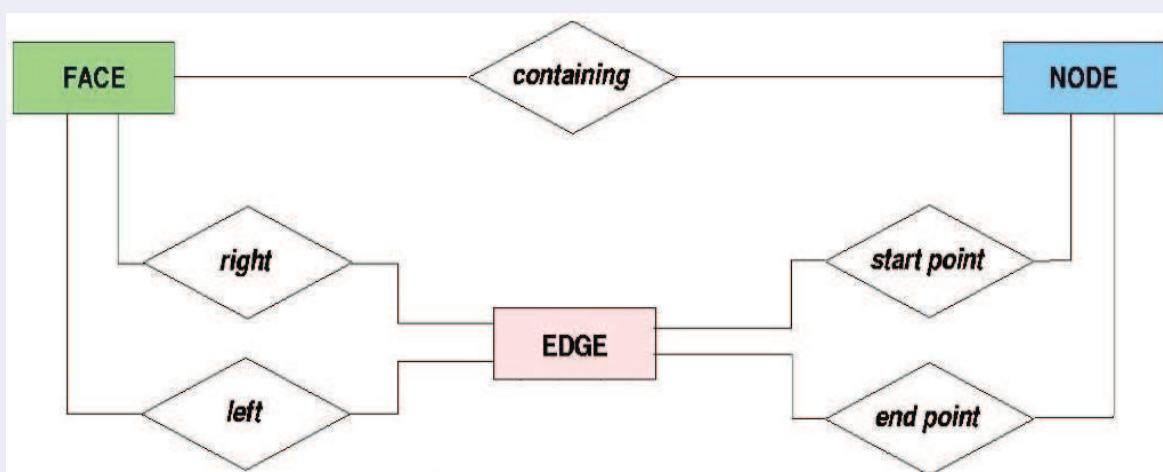
## Orden de almacenamiento de celdas

Morton es similar a quadtrees

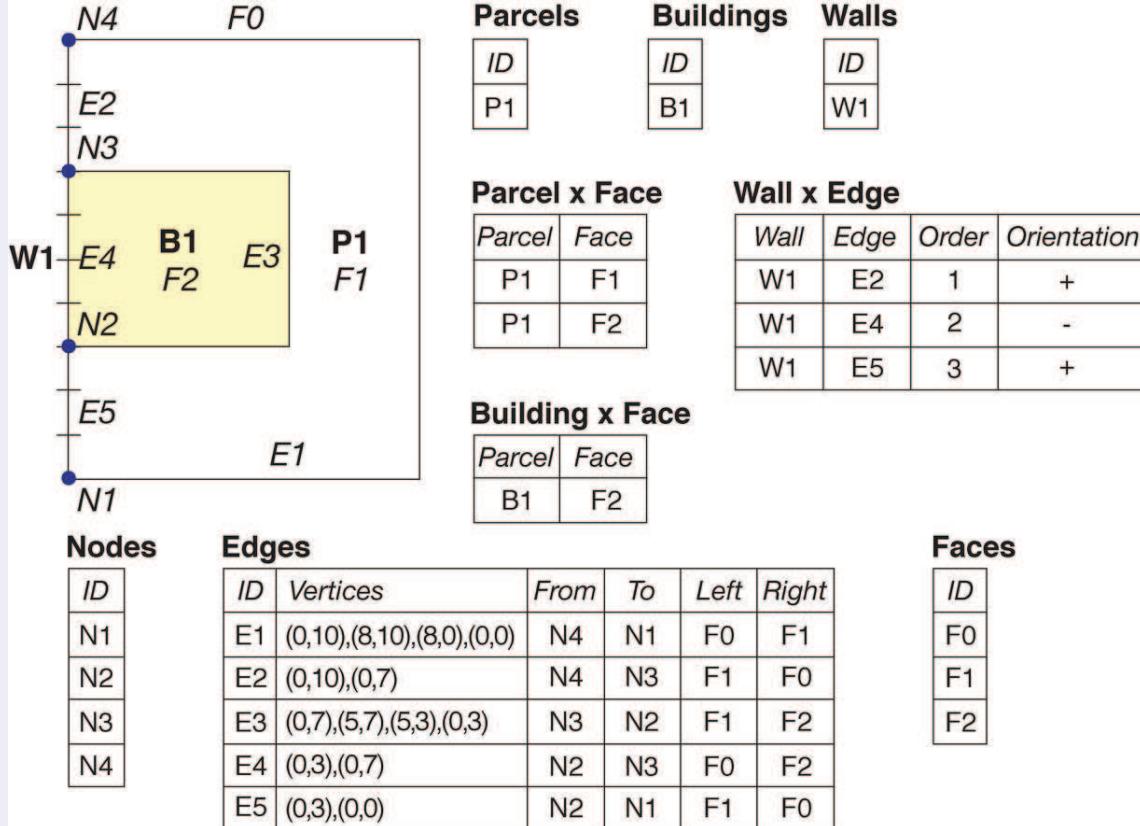


# Modelo Topológico

## Esquema conceptual del Modelo Topológico



# Ejemplo de Modelo Topológico



## Conclusiones

- 1 Uso de Bases de Datos Espaciales
- 2 Diseño de esquemas
- 3 Representación Geográfica y Nivel Lógico
- 4 Nivel Físico
- 5 Conclusiones
- 6 Bibliografía

# Conclusiones

- Los SGBD son una parte esencial de GIS.
- Aportan una forma estándar de almacenar y acceder a los datos.
- Los SGBD de ámbito general se han extendido para soportar Información Geográfica.
- Se basan en estándares: SQL, OGC.

## Bibliografía

- 1 Uso de Bases de Datos Espaciales
- 2 Diseño de esquemas
- 3 Representación Geográfica y Nivel Lógico
- 4 Nivel Físico
- 5 Conclusiones
- 6 Bibliografía

# Bibliografía

- BML15 P.A Burrough, R.A. McDonnell, and C.D. Lloyd. *Principles of Geographical Information Systems*. Oxford University Press, 2015.
- Her11 John R. Herring (Ed.). *OpenGIS Implementation Standard for Geographic Information - Simple feature access - Part 1: Common architecture*. Open Geospatial Consortium Inc., 2011.
- LGMR15 Paul A. Longley, Michel F. Goodchild, David J. Maguire, and David W. Rhind. *Geographic Information Science and Systems (Fourth Edition)*. Wiley, 2015.
- Mar15 Angel Marquez. *PostGIS Essentials*. Packt Publishing, 2015.
- OH15 Regina O. Obe and Leo S. Hsu. *PostGIS in Action (Second Edition)*. Manning, 2015.
- WD04 Michael Worboys and Matt Duckham. *GIS A Computing Perspective (Second Edition)*. CRC Press, 2004.
- Zei99 Michael Zeiler. *Modeling Our World*. ESRI Press, 1999.