

# BASES DE DATOS GEOGRÁFICAS

## *OGC*

“Lo más grande es el espacio, porque lo encierra todo”  
Tales de Mileto



- Docentes :     Ing. Fernando Sato  
                         A.S. Sebastian Trossero

# RESUMEN

---

## Introducción a las Ciencias de Información Geográfica:

- Conceptos sobre GIS (Sist.Inf.Geog.)
- Conceptos de BD Geograficas
- Elementos Básicos de Análisis Espacial
- Elementos de cartografía temática
- Extensiones Espaciales de Servidores Comerciales
- Ejercicio Propuesto

# Bases de Datos Geográficas

## Motivación y Requisitos Funcionales

---

El propósito de un SIG es la gestión integrada de datos espaciales y temáticos. Las bases de datos relacionales deben sumar requisitos adicionales tales como:

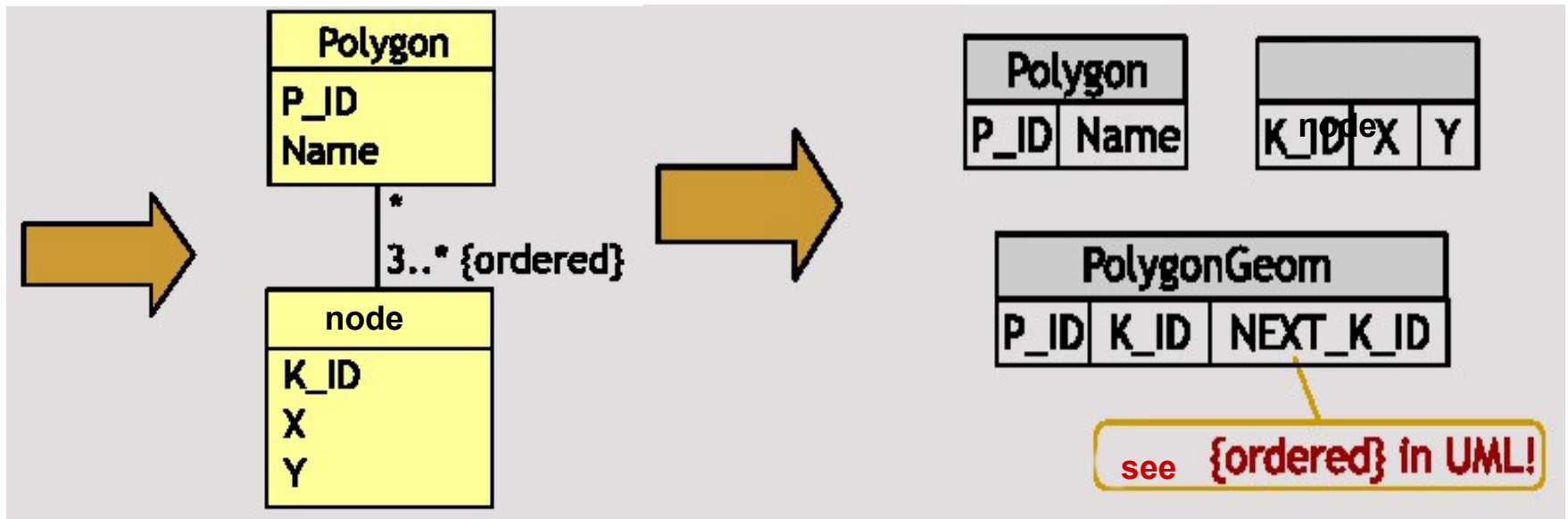
- ✓ **Representación de datos espaciales:** Toda forma de datos espacial relevante para la aplicación debe ser almacenable en el sistema, ej. Límites de un edificio.
- ✓ **Consultas espaciales:** El sistema de información debe soportar queries con contexto o referencia espacial.  
ej. ¿Qué propiedades se encuentran al lado de una carretera principal?
- ✓ **Integración con datos temáticos:** los usuarios deben tener la capacidad de combinar datos temáticos y espaciales en una forma adecuada.
- ✓ ej. Asociar los límites de propiedad con el propietario, las características de construcción, impuestos, etc.

# Bases de Datos Geográficas

## Intento de Solución

Diseño de base de datos para un SIG, las estructuras de datos espaciales pueden ser modeladas como estructuras de datos temáticos (no espaciales):

Ejemplo: modelado de un tipo para polígonos con gestión de punto no redundante.



*Opinión: ¿Parece razonable?*

# Bases de Datos Geográficas

## Motivación / Solución

---

La solución parece factible técnicamente, pero operativamente?:

- ✓ ¿Será tan eficiente como tener un soporte "nativo"?
- ✓ ¿Hay posibilidad de usar índices para queries espaciales?
- ✓ ¡Las operaciones y los predicados deben evaluarse fuera del SGBD!
- ✓ **Violación de la independencia de datos:** la reorganización de la estructura de almacenamiento requiere cambios en los programas de aplicación.

# Bases de Datos Geográficas

## Motivación / Solución

---

Todo lleva a pensar que una mejora significativa sería usar tipos de datos espaciales específicos, que se implementan como los llamados tipos de datos abstractos (**TDA**) dentro del DBMS.

- ✓ Uso en analogía a tipos de datos SQL primitivos
- ✓ Es posible utilizar estructuras de índices espaciales dentro del proceso de consulta.
- ✓ Los operadores espaciales y predicados pueden ser evaluados utilizando algoritmos especiales dentro del SGBD.

# Bases de Datos Geográficas

## Motivación / Solución

---

Un tipo de datos abstracto (TDA) es una representación independiente de la implementación de un conjunto de dominios de valor y operaciones para formar una unidad.

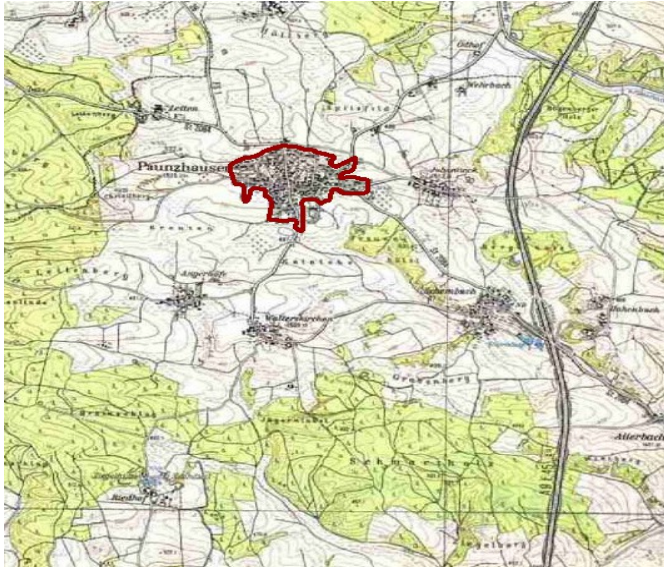
En nuestro contexto, las TDA tienen las siguientes características deseables:

- ✓ Los tipos de datos complejos son más fáciles de usar: por ejemplo, un tipo de polígono puede usarse para especificar el dominio de valor para una columna en un esquema relacional
- ✓ **Encapsulación:** La estructura de datos interna está oculta al usuario
- ✓ **Interfaz:** El acceso desde el exterior se limita a operaciones predefinidas
- ✓ La interfaz y la implementación están **separadas** (a nivel conceptual). *Ideal para no violar la independencia de datos.*


# Bases de Datos Geográficas

## Motivación / Solución

---



```
CREATE TABLE cities  
( name      VARCHAR(30) ,  
  residents INTEGER ,  
  geometry  POLYGON )
```

cities		
name	residents	geometry
Paunzhausen	8500	
...	...	...

La estructura de datos real del tipo de datos abstractos POLYGON se oculta al exterior.

**Consulta:** ¿Puede ser posible una solución propietaria?



# Bases de Datos Geográficas

## Motivación / Solución

---

Motivación → Requisitos Funcionales:

- ✓ Representación de datos espaciales.
- ✓ Consultas espaciales.
- ✓ Integración con datos temáticos.

Requisitos NO Funcionales:

- ✓ Eficiencia y Performance.

**Nota:** Imaginemos operaciones complejas del tipo de la sig, con restricciones de miles de objetos.

*Asociar los límites de propiedad con el propietario, las características de construcción, impuestos, etc.*

**Solución:** *Extender SQL con funcionalidad espacial → Mas Standard*

# Bases de Datos Geográficas

## Open Geospatial Consortium

---

El **Open Geospatial Consortium** (OGC) es un consorcio internacional que agrupa:

- ✓ Empresas
- ✓ Universidades
- ✓ Organismos Estatales

Con las siguientes metas fundamentales:

- ✓ Proveer de estándares abiertos, gratuitos y públicos.
- ✓ Liderar la creación de estándares que permitan que el contenido y los servicios GeoEspaciales se integren a procesos cívicos y de negocios.
- ✓ Facilitar la adopción de arquitecturas de referencia abiertas en materia de información espacial.

# Bases de Datos Geográficas

## OGC Simple Features Standard

---

- OGC define un modelo de datos para objetos espaciales conocido como Simple Features Standard (**SFS**) (Estandar ISO 19115).
- El estándar se divide en dos partes:
  - Arquitectura común (Modelo de Objetos - datos)
  - Implementación en SQL (Permite construir bases de datos geográficas)

# Bases de Datos Geográficas

## OGC Simple Features Standard

---

El Open Geospatial Consortium (OGC), entre otras cosas, ofrece la especificación de características simples para SQL:

- ✓ Describe un conjunto de tipos de datos de geometría para SQL basado en el modelo de geometría propio (OGC).
- ✓ Describe un conjunto de operaciones SQL para estos tipos.

No describe cuestiones físicas como índices, implementación interna, rangos de valores, etc.

# Bases de Datos Geográficas

## OGC Simple Features Standard

---

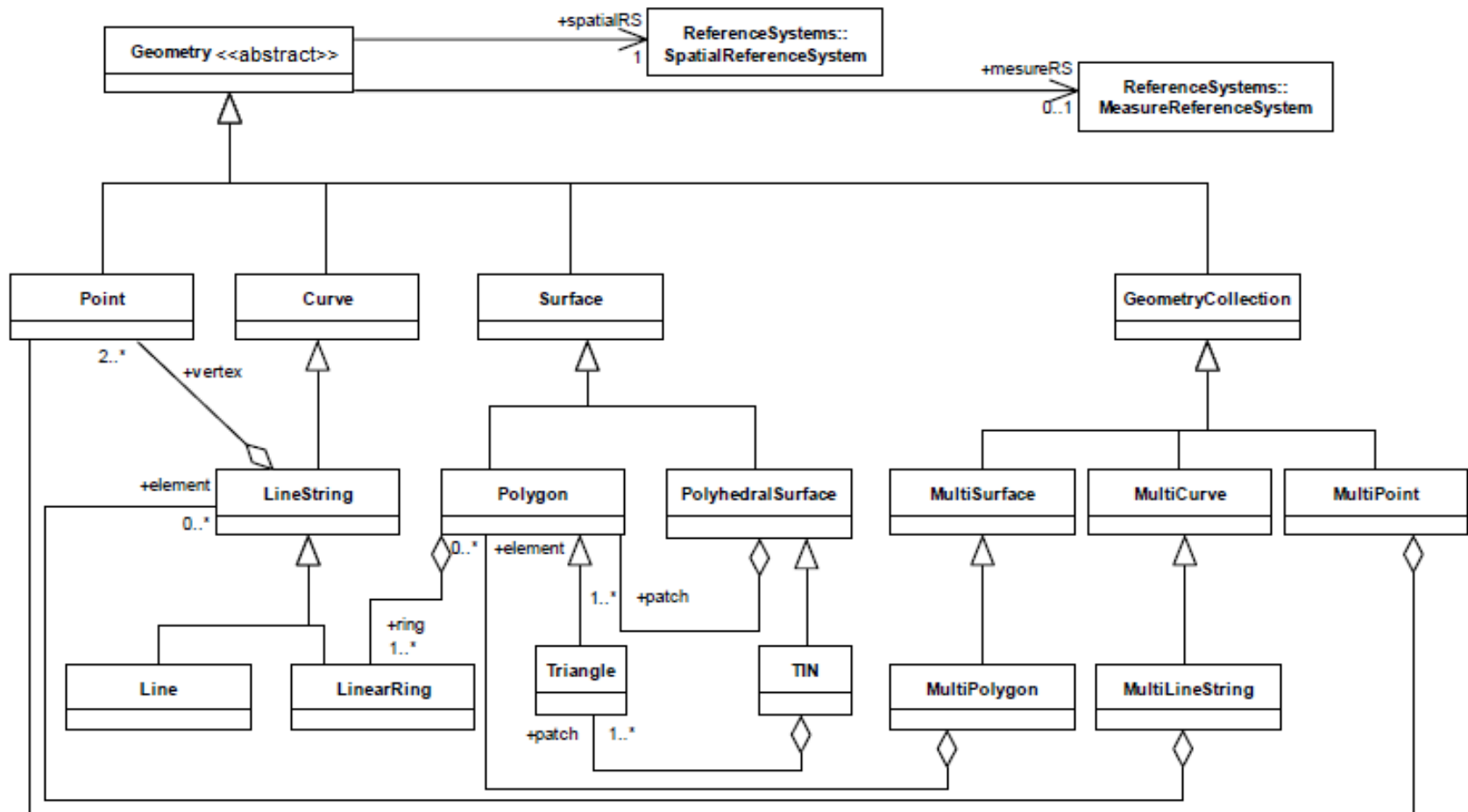
### Características Espaciales:

- ✓ Término "característica": abstracción de un fenómeno del mundo real ("geoObjeto");  
Una característica se almacena como un dato simple o un dato complejo (conjunto de datos).
- ✓ Modelado de la geometría de objetos espaciales:
  - ✓ Mínimo 2 dimensiones espaciales, admiten también z y m (3 y 4): X, Y, Z referencian un punto en el espacio, m es otra característica asociada al punto, ejemplo: temperatura, ejemplo: área de servicios en la ruta, ejemplo: mojón km.
  - ✓ Sólo interpolación lineal entre puntos
  - ✓ No hay representación explícita de la topología

# Bases de Datos Geográficas

## OGC - Modelo de datos

### Diagrama de Clases: Tipos agregados

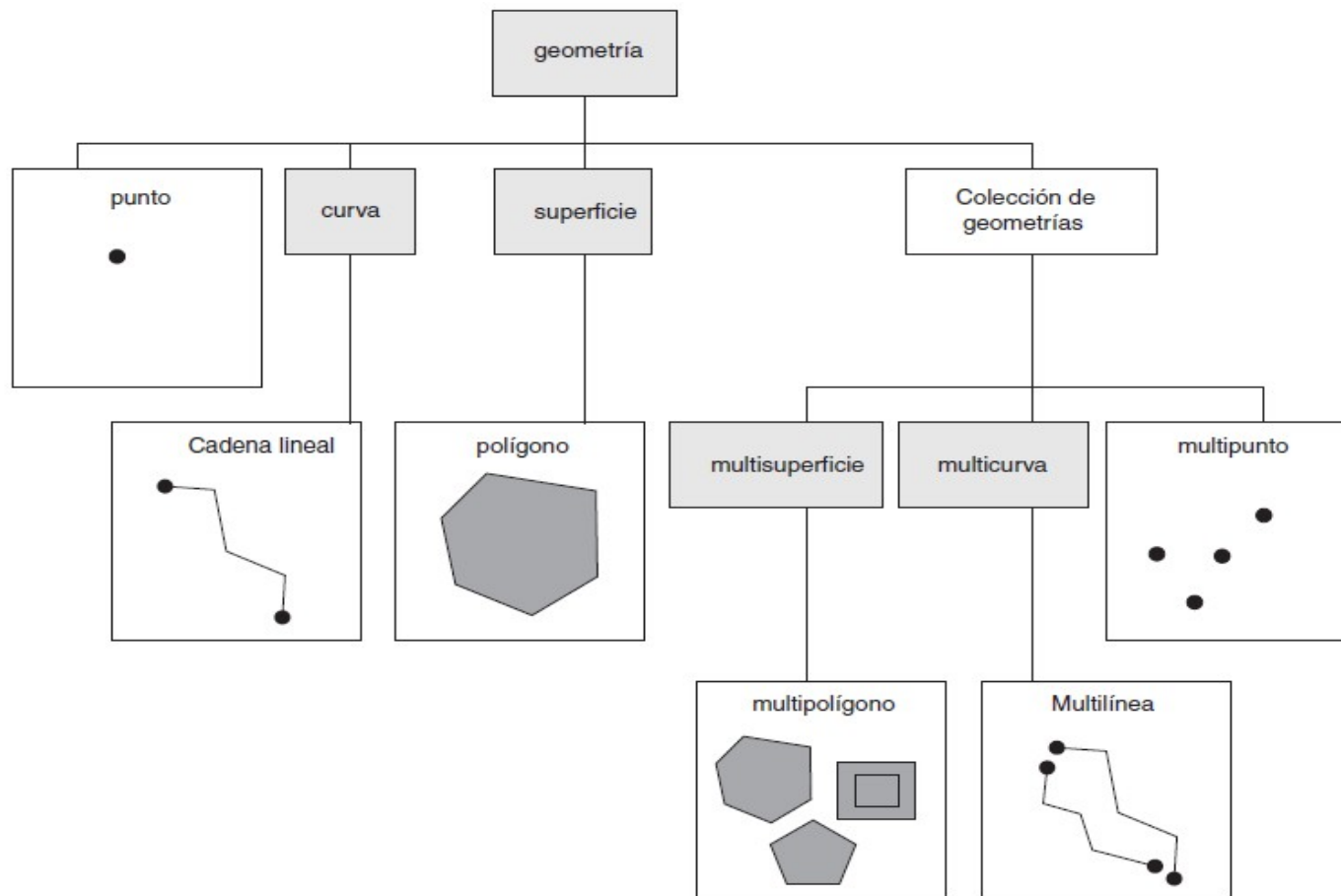


# Bases de Datos Geográficas

## OGC - Modelo de datos

---

### Diagrama de Conceptual: Tipos mas usados



# Bases de Datos Geográficas

## OGC – Características de Geometry

---

### **Características Geometry:**

- Es la superclase de toda la jerarquía; Es abstracto.
- Soporta diferentes operaciones en las siguientes áreas:
  - Operaciones básicas
  - Pruebas de relaciones topológicas
  - Operaciones espaciales
- A cada objeto de este tipo se le asigna un sistema de referencia espacial (`class SpatialReferenceSystem`).
- La subclase `GeometryCollection` representa conjuntos de objetos `geometry` simples. Tales conjuntos también pertenecen *geometry*.
  - Todos los elementos de una `GeometryCollection` deben referirse a un sistema de referencia espacial común.



# Bases de Datos Geográficas

## OGC – Operaciones de Geometry

---

### **Operaciones Básicas sobre Geometry – Características:**

La especificación OGC define las siguientes operaciones básicas (G representa un `geometry`, una geometría).

- ✓ **Dimension(g)**: devuelve la dimensión de la geometría g

Si la geometría proporcionada está **vacía**, debe devolver el valor **-1**. Para **puntos** y **multipuntos**, la dimensión es **cero 0** (cero); para curvas y multicurvas, la dimensión es **1**; y para **polígonos** y **multipolígonos**, la dimensión es **2**. Si la geometría proporcionada es un **valor nulo**, se devuelve un valor nulo.

- ✓ **GeometryType(g)**: devuelve el nombre del tipo geométrico de g (por ejemplo, LINESTRING, POLYGON, MULTICURVE, ...).
- ✓ **SRID(g)**: devuelve el ID del sistema de referencia espacial de g

# Bases de Datos Geográficas

## OGC – Operaciones de Geometry

---

### **Operaciones Básicas sobre Geometry – Análisis:**

La especificación OGC define las siguientes operaciones básicas (G representa un `geometry`, una geometría).

- ✓ **IsEmpty(g)**: pruebas, si g está vacío.
- ✓ **IsSimple(g)**: tests, si g es simple (definido en el modelo de geometría OGC).

# Bases de Datos Geográficas

## OGC – Operaciones de Geometry

---

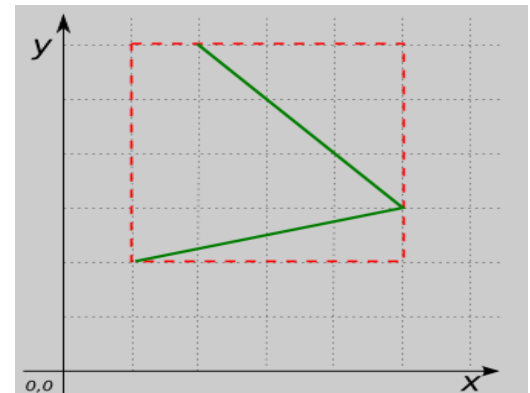
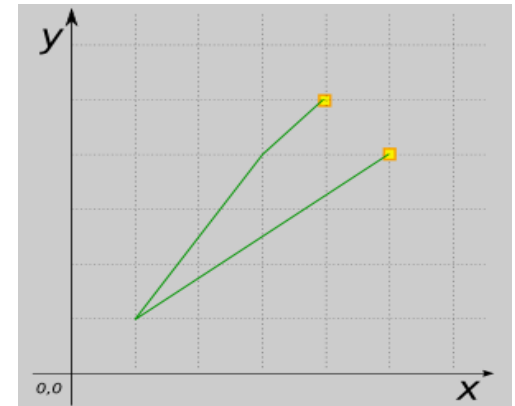
### Operaciones Básicas sobre Geometry – Análisis:

La especificación OGC define las siguientes operaciones básicas (G representa un `geometry`, una geometría).

- ✓ **Boundary(g)**: genera un contorno común de un objeto.

Caso particular Input 'LINESTRING(4 6, 1 1, 5 5)'  
Salida Boundary: MULTIPOINT((4 6), (5 5))

- ✓ **Envelope(g)**: genera un recuadro mínimo, rectángulo paralelo al eje alrededor de un objeto
- ✓ ("minimum bounding rectangle", MBR).



# Bases de Datos Geográficas

## OGC – Operaciones de Geometry

---

### **Operaciones Básicas sobre Geometry – Representación:**

La especificación OGC define las siguientes operaciones básicas (G representa un `geometry`, una geometría).

- ✓ **AsText(g)**: convierte la geometría g en el "formato de texto SQL" para exportar a otras aplicaciones.

por ejemplo: **POLYGON (0 0,0 1,1 1,1 0,0 0))**

- ✓ **AsBinary(g)**: convierte g en formato binario (también especificado) para exportarlo a otras aplicaciones

✓

# Bases de Datos Geográficas

## OGC – Point

---

### Elemento Básico → POINT:

- ✓ Objeto geométrico 0D (punto).
- ✓ Coordenadas x,y (z,m opcionales).
- ✓ Su limite es el conjunto vacío.
- ✓ Su **envelope** es el propio punto.
- ✓ Posee métodos para obtener sus coordenadas (funciones x e y)

# Bases de Datos Geográficas

## OGC – Curve

---

### Elemento Básico → CURVE:

- ✓ Curve (abstracto): representa la curva definida por una secuencia de puntos.
- ✓ Objeto geométrico 1D (punto).
- ✓ Métodos:
  - ✓ length: devuelve el largo.
  - ✓ Startpoint y endpoint: punto inicial y final.
  - ✓ Isclosed: devuelve 1 si `startpoint = endpoint`.
  - ✓ Isring: devuelve 1 si `isclosed` y no pasa mas de una vez por el mismo punto.
- ✓ El limite de una curva cerrada (*isring*) es el conjunto vacio.
- ✓ El limite de una curva abierta son el `startpoint` y el `endpoint`.

# Bases de Datos Geográficas

## OGC – LineString

---

### **Elemento Básico → LINESTRING:**

- ✓ Es una curva con interpolación lineal entre los puntos.
- ✓ Su envelope es un polígono, o si es una línea vertical u horizontal, es la propia línea.
- ✓ Métodos:
  - ✓ numPoints(): devuelve cantidad de puntos
  - ✓ pointN(linestring, n): devuelve el punto n
- ✓ Line
  - ✓ Es un LineString de 2 puntos (un segmento de recta)
- ✓ LinearRing
  - ✓ En un LineString simple y cerrado (un anillo)

# Bases de Datos Geográficas

## OGC – Surface

---

### **Elemento Básico → SURFACE:**

- ✓ Surface (abstracta)
- ✓ Objeto geométrico 2D que representa una superficie.
- ✓ Métodos:
  - ✓ `area()`: devuelve el área en el SRS correspondiente.
  - ✓ `centroid()`: devuelve el centroide (baricentro) de la superficie (puede ser un punto exterior)
  - ✓ `pointOnSurface()`: devuelve un punto cualquiera perteneciente a la superficie



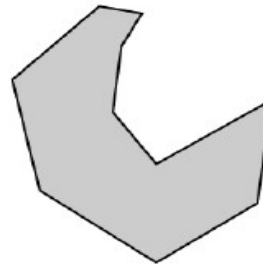
# Bases de Datos Geográficas

## OGC – Polygon

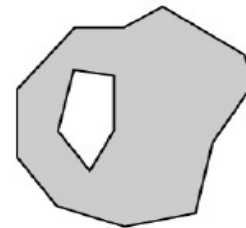
---

### Elemento Básico → POLYGON:

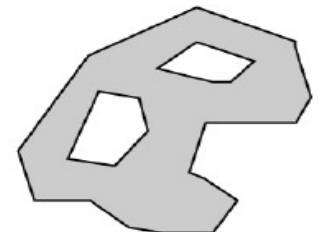
- ✓ Es una superficie plana definida por un LinearRing exterior y 0 o más LinearRings interiores, para permitir polígonos con huecos.
- ✓ Los polígonos son objetos simples: no existe intersección entre sus contornos.
- ✓ Métodos:
  - ✓ exteriorRing(): devuelve el anillo exterior (linestring)
  - ✓ numInteriorRings(): devuelve la cantidad de anillos interiores.
  - ✓ interiorRingN(): devuelve el anillo interior N
- ✓ El limite de un polígono son su anillo exterior y sus anillos interiores.



a)



b)



c)

# Bases de Datos Geográficas

## OGC – Operaciones Topológicas

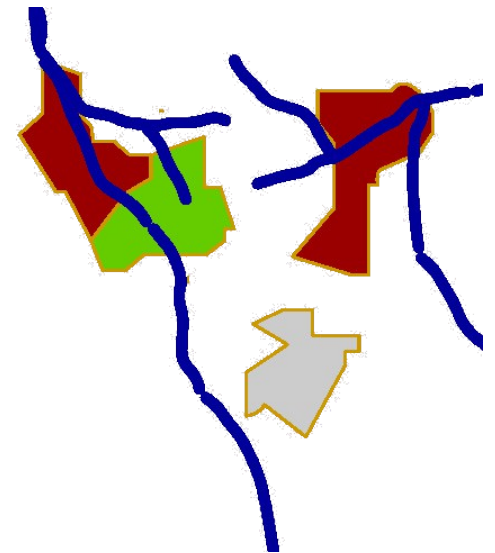
---

En un entorno SQL OGC compatible, tenemos la capacidad de evaluar relaciones topológicas.

Ejemplo:

¿Qué ciudades son atravesadas por Ríos?

```
SELECT DISTINCT Ciudades.name  
FROM Ciudades JOIN Rios ON  
    CROSSES (Ciudades.Geom, Rios.Geom) = 1
```



Las operaciones relacionadas con las relaciones topológicas se basan en el denominado "modelo de intersección extendido de nueve dimensiones" y están directamente disponibles en consultas SQL del tipo del ejemplo.

# Bases de Datos Geográficas

## OGC – Operaciones Topologicas

---

### Operaciones de Relaciones Topologicas:

Los parámetros g1 y g2 son en cada caso dos geometrías, el valor de retorno es un entero: 0 false , 1 true, -1 o NULL si es desconocido, significa que uno de los argumentos contiene valor NULL:

- ✓ **Equals(g1, g2):** g1 y g2 son espacialmente idénticos.
- ✓ **Disjoint (g1, g2):**  $\leftrightarrow$  si g1 y g2 son espacialmente disjuntos.
- ✓ **Touches(g1, g2):** Los límites de g1 y g2 se tocan pero no su interior.
- ✓ **Crosses(g1,g2):** La dimensión de la intersección de g1 y g2 es menor que la dimensión máxima de g1 y g2 y la intersección comprende los puntos interiores de g1 y g2.
- ✓ **Within(g1, g2)/Contains(g2, g1):** g1 está completamente en g2
- ✓ **Overlaps(g1, g2):** g1 esta superpuesto con g2 .

# Bases de Datos Geográficas

## OGC – Operaciones Espaciales

---

### **Operadores Espaciales:**

Provenientes de teoría de conjuntos.

- ✓ **Intersection(g1, g2):** Genera la geometría intersección.
- ✓ **Union(g1,g2):** Genera la geometría union entre g1 y g2.
- ✓ **Difference(g1, g2):** Genera la geometría diferencia entre g1 y g2.

# Bases de Datos Geográficas

## OGC – Operaciones Espaciales

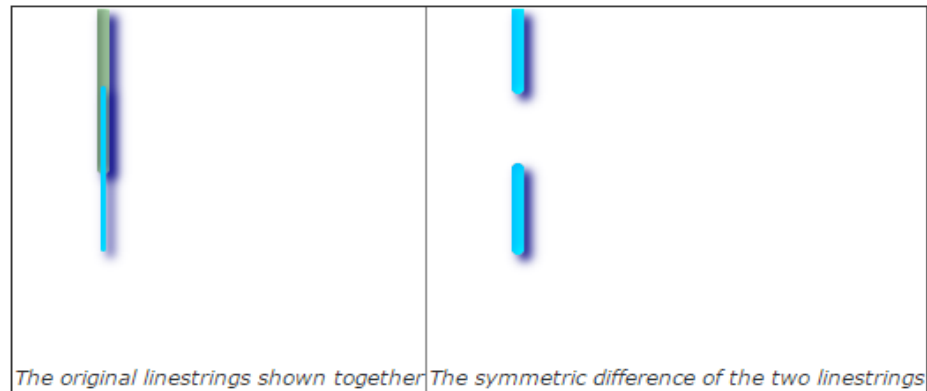
---

### Operadores Espaciales:

- ✓ **SymDifference(g1, g2)**: Devuelve una geometría que representa las porciones de A y B que no se intersectan. Se llama diferencia simétrica porque  $\text{SymDifference}(A, B) = \text{SymDifference}(B, A)$ . Uno puede pensar en esto como:

Union (geomA, geomB)  
- Intersection (A, B).

### Examples



```
--Safe for 2d - symmetric difference of 2 linestrings
SELECT ST_AsText(
    ST_SymDifference(
        ST_GeomFromText('LINESTRING(50 100, 50 200)'),
        ST_GeomFromText('LINESTRING(50 50, 50 150)')
    )
);

st_astext
-----
MULTILINESTRING((50 150,50 200),(50 50,50 100))
```

# Bases de Datos Geográficas

## OGC – Operaciones Espaciales

---

### Operadores Espaciales:

- ✓ **Distance(g1, g2):** Genera la distancia más corta entre dos geometrías en el sistema de referencia correspondiente.

### Operadores Espaciales que generan geometrías:

- ✓ **Buffer(g, d):** Genera una geometría (Polygon) con una distancia d a cada punto del exterior de g.
- ✓ **ConvexHull(g):** Genera la geometría que es el casco convexo de g.



# Bases de Datos Geográficas

## Tipos Geometricos y Formatos: WKT

---

OGC proporciona un formato de texto, denominado "Well-known text representation (**WKT** representación de texto bien conocido)" para el intercambio de datos y para la construcción de objetos geométricos.

Dicho de otra forma, **WKT** es una descripción textual de la geometría de un objeto, por Ej:

Geometric type	WKT	Detalle
Point	<code>POINT (10 10)</code>	<i>Un punto</i>
LineString	<code>LINESTRING (10 10,20 20,30 40)</code>	<i>Un LineString con 3 puntos</i>
Polygon	<code>POLYGON ((10 10,10 20, 20 20,20 15,10 10))</code>	<i>Un Polígono</i>
Multipoint	<code>MULTIPOINT (10 10, 20 20)</code>	<i>Un MultiPoint con 2 puntos</i>
MultiLineString	<code>MULTILINESTRING ((10 10, 20 20), (15 15 ,30 15))</code>	<i>A MultiLineString with 2 LineStrings</i>
Geometry Collection	<code>GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20))</code>	<i>Una GeometryCollection compuesto por 2 puntos y un LineString</i>

# Tipos Geometricos y Formatos: WKB

### Características:

- ✓ Ej:

	st_asewkb bytea
1	\001\001\000\000\000\000\000\000\000\000\000\000\360?\000\000\000\000\000\000\000



# Bases de Datos Geográficas

## Tipos Geometricos y Formatos de Texto


Como vimos la función **AsText** permite ver la representación WKT de los objetos gráficos.

Por Ejemplo: Si realizamos el siguiente query.

```
SELECT Nombre, ATEXT (Geom) as Geometry  
FROM Ciudades  
WHERE Nombre = "Paunzhausen"
```

Resultado:

```
CREATE TABLE Cities  
( Name VARCHAR(30) ,  
Residents INTEGER ,  
Geometry POLYGON )
```

Cities		
Name	Residents	Geometry
Paunzhausen	8500	

Name	Geometry
Paunzhausen	POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10))

# Bases de Datos Geográficas

## Tipos Geometricos y Formatos de Texto

---

También especifica una función `GeoFromText`, que recibe como argumento una especificación WKT y genera el tipo específico descripto.

Sintaxis:

```
geometry GeomFromText(text WKT)
geometry GeomFromText(text WKT, integer srid)
```

Hay dos variantes de la función `ST_GeomFromText`. El primero no toma SRID y devuelve una geometría sin sistema de referencia espacial definido (`SRID = 0`). El segundo toma un SRID como el segundo argumento y devuelve una geometría que incluye este SRID como parte de sus metadatos.

Ejemplo:

```
GeomFromText('TRIANGLE((0 0, 10 0, 0 10, 0 0))')
```

Genera un objeto Triangle.

**Nota:** También especifica constructores equivalentes para tipos concretos, ej: `PolygonFromText`, etc.

# Bases de Datos Geográficas

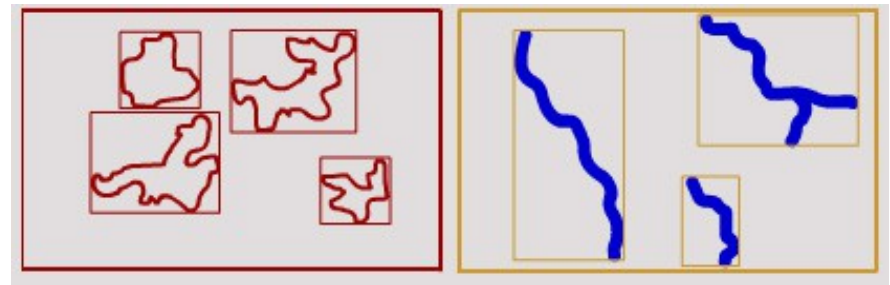
## Consultas Típicas

---

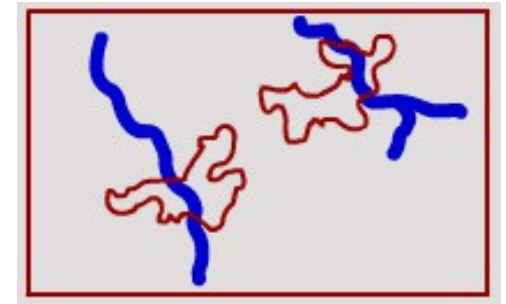
Uno de los tipos de consulta más importantes en bases de datos son los join o reunión. Una reunión espacial es una combinación "general", que comprende un predicado espacial dentro de la condición de consulta.

Ejemplo:

¿Que ríos pasan por ciudades?



```
SELECT Cities.Name, Rivers.Name  
FROM Cities JOIN Rivers ON  
crosses(Cities.Geometry,Rivers.Geometry)=1
```



**Espacio de discusión:**

¿Que diferencia una reunión espacial con una sql92?

# Bases de Datos Geográficas

## Consultas Típicas

---

Otro Ejemplo:

Ejemplo:

¿Qué Ríos se encuentran (completamente) en argentina?

```
SELECT Rivers.*  
FROM Rivers JOIN Countrys ON  
    Countrys.Name='Argentina'  
AND WITHIN(Rivers.Geometry, Country.Geometry)=1
```

**Espacio de discusión:**

¿Que devuelve la consulta?

# Bases de Datos Geográficas

## Fuentes

---

En general:

<http://www.opengeospatial.org>

En particular:

<http://www.opengeospatial.org/standards/sfs>