

Bases de Datos Espaciales

Mariano Beiró

Dpto. de Computación - Facultad de Ingeniería (UBA)

14 de noviembre de 2018

Topics

- 1 Introducción
- 2 Simple Features
- 3 Otros formatos vectoriales
 - Shapefiles
 - GeoJSON
 - KML y GML
- 4 Indexado de datos espaciales
- 5 Ejercicio en PostGIS
- 6 Bibliografía

1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles
- GeoJSON
- KML y GML

4 Indexado de datos espaciales

5 Ejercicio en PostGIS

6 Bibliografía

Introducción

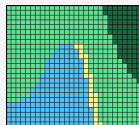
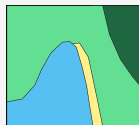
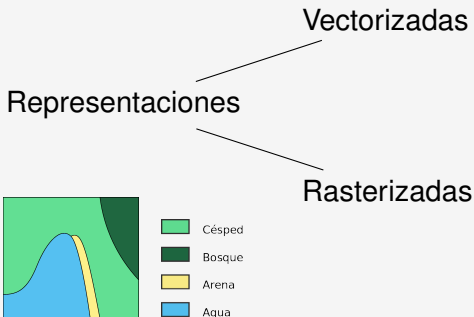
Bases de Datos Espaciales y GIS

[ELM16 26.3.1]

- Las **bases de datos espaciales** permiten representar en forma eficiente objetos definidos en un espacio geométrico.
 - Objetos simples: puntos, líneas, polígonos
 - Objetos más complejos: objetos 3D, teselados
- Son parte de un contexto más general : los **sistemas de información geográfica** (GIS), que permiten:
 - Almacenar y manipular datos geográficos (función de base de datos)
 - Capturar, analizar y presentar visualmente datos geográficos (funciones más generales de los sistemas de información)

Introducción

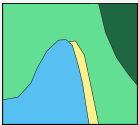
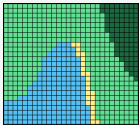
Tipos de representación



- En las representaciones **vectorizadas**, los objetos se representan a través de vectores que identifican sus bordes.
- En las representaciones **rasterizadas**, los objetos se proyectan sobre una matriz de celdas (grilla).

Introducción

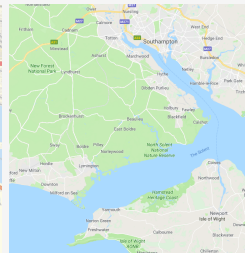
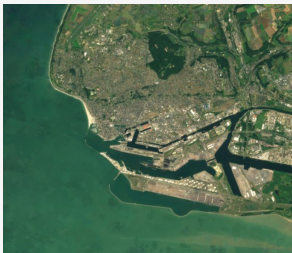
Vector vs. Raster

	Vector	Raster
		
Estándares (Ejemplos)	Simple Features GeoJSON Shapefile	GeoTIFF JPEG 2000 NetCDF
Ventajas	Alto nivel de detalle. Fácil de mantener y actualizar.	Representación de datos continuos (ej., elevación). Fácil hacer análisis cuantitativo. Fácil de renderizar.
Desventajas	Pobre representación de datos continuos.	El nivel de detalle depende de la resolución. Suelen ocupar más espacio de almacenamiento.

Introducción

Basemaps

- Los formatos raster también se utilizan para almacenar en un GIS el **mapa de fondo (*basemap*)** sobre el que luego se muestran otras capas vectoriales o rasterizadas.
- Estos *basemaps* suelen obtenerse de fotografías aéreas, satelitales, o mapas escaneados.
- A veces ya contienen información renderizada sobre ellos.



1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles
- GeoJSON
- KML y GML

4 Indexado de datos espaciales

5 Ejercicio en PostGIS

6 Bibliografía

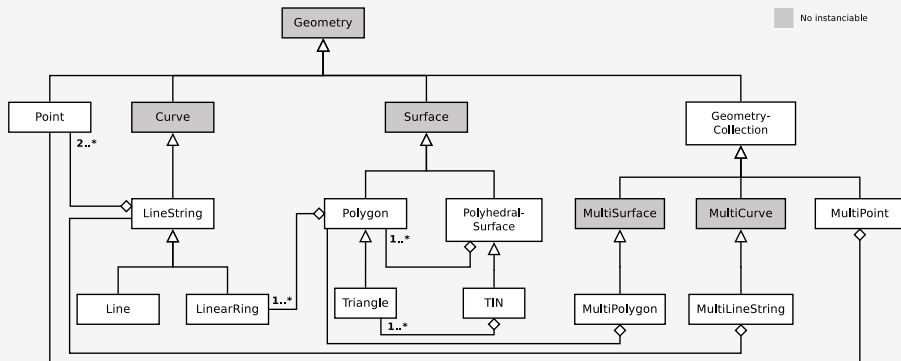
Simple Features

- Fue propuesto como **estándar** por el Open Geospatial Consortium (OGC) de 1997.
- Actualmente es también estándar ISO (**ISO 19125:2004**).
- Define cómo agregar funcionalidad espacial a las bases de datos a través de una representación vectorizada, especificando con notación UML:

Geometrías	Puntos, líneas, polígonos, etc.
Métodos básicos	Métodos asociados al objeto geométrico. Ej.: longitud de una línea, área de un polígono.
Predicados de relación	Métodos que verifican una relación topológica entre el objeto y otro. Ej.: si dos polígonos se intersectan, si un punto está dentro de un polígono.
Mét. de análisis espacial	Son métodos que permiten operar al objeto con otro. Ej.: intersectar polígonos, intersectar una línea con un polígono.





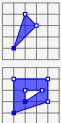
Simple Features

Geometrías






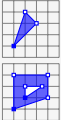
Simple Features

Geometrías

Ejemplo	Nombre	Significado
	Point	Un objeto de dimensión 0 en un sistema de coordenadas. Tiene coordenadas x e y . Puede tener coordenadas z y m (medida) también.
	Curve	(Clase abstracta) Una secuencia de puntos con un tipo de interpolación entre ellos.
	LineString	Una curva con interpolación lineal entre puntos. Es el único tipo de <i>Curve</i> definido en el estándar.
	Line	Un LineString con exactamente dos puntos.
	Polygon	Una superficie planar definida por un borde exterior (<i>LineString</i>) y 0 ó más bordes interiores. Generalización del polígono simple, denominada en matemática <i>weakly simple polygon</i> .




Simple Features

Métodos Básicos

Ejemplo	Nombre	Significado
	Point	<code>X() : Double</code> <code>Y() : Double</code>
	Curve	<code>Length() : Double</code> <code>StartPoint() : Point</code> <code>IsClosed() : Integer</code>
	LineString	<code>NumPoints() : Integer</code> <code>PointN(N: Integer) : Point</code>
	Polygon	<code>ExteriorRing() : LineString</code> <code>Area() : Double</code> <code>Centroid() : Point</code>

Simple Features

Predicados de relación

Nombre	Ejemplo	Resultado
<code>G.Disjoint(G2:Geometry):Integer</code>		0
<code>G.Intersects(G2:Geometry):Integer</code>		1
<code>G.Contains(G2:Geometry):Integer</code>		0
<code>G.Within(G2:Geometry):Integer</code>		1
<code>G.Within(G2:Geometry):Integer</code>		0

Simple Features




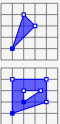
Métodos de análisis espacial

Nombre	Ejemplo	Resultado
<code>G.Distance (G2:Geometry) :Double</code>		$\sqrt{2}$
<code>G.Intersection (G2:Geometry) :Geometry</code>		
<code>G.Difference (G2:Geometry) :Geometry</code>		

Simple Features

Well-known Text Representation (WKT)

- Simple Features define una representación **textual** estándar para sus objetos geométricos, denominada **Well-known Text Representation (WKT)**.

Ejemplo	Geometría	WKT
	Point	POINT (2 3)
	LineString	LINESTRING (1 4, 3 2, 2 1)
	Line	LINE (1 4, 3 2)
	Polygon	POLYGON ((1 1, 2 4, 3 3)) POLYGON ((1 1, 1 4, 4 4, 4 2), (2 2, 3 3, 2 3))

Simple Features

Well-known Binary Representation (WKB)

- Se define también una representación **binaria** estándar, conocida como **(WKB) *Well-known Binary Representation***.
- Esta representación puede ser utilizada para transmitir objetos geográficos como *streams* de bytes (por ejemplo, entre un cliente ODBC y una base de datos relacional), o para almacenarlos en forma binaria en una base de datos.
- WKB es una representación **portable**:
 - Convierte cada geometría en una secuencia de datos numéricos de tipo **unsigned integer** (32 bits) ó **double** (IEEE 754, 64 bits).
 - Se puede optar por codificar los números en:
 - XDR: eXternal Data Representation (*big endian*)
 - NDR: Network Data Representation (*little endian*)
 - La codificación elegida se indica con un tag de un byte que precede a la serialización.

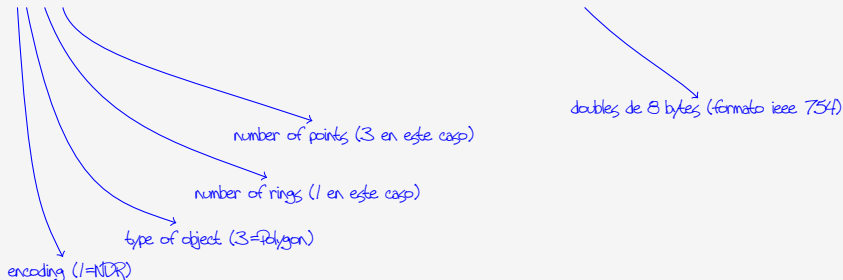
Simple Features

Well-known Binary Representation (WKB): Ejemplo



POLYGON (1 1, 2 4, 3 3)

1	3	1	3	1	1	2	4	3	3
				X1	Y1	X2	Y2	X3	Y3



Simple Features

Implementaciones SQL

- El estándar también especifica 2 formas posibles de implementar su funcionalidad en SQL.
 - Basada en la definición de tipos de dato geométricos
 - Basada en tipos de datos predefinidos
- En la implementación con **tipos de dato geométricos** se definen:
 - Nuevos tipos de dato ¹
 - ST_Point, ST_Line, ST_Polygon, ...
 - Funciones
 - ST_Equals(geometry g), ST_Contains(geometry g), ST_Length(geometry g), ...
 - Conversión de WKT a tipo
 - ST_LineFromText(text g, integer srid), ST_PolyFromText(text g, integer srid), ...
 - Conversión de WKB a tipo
 - ST_LineFromWKB(bytea g, integer srid), ST_PolyFromWKB(bytea g, integer srid), ...
 - Conversión de tipo a WKT
 - ST_AsText(geometry g)
 - Conversión de tipo a WKB
 - ST_AsBinary(geometry g)

¹ La notación ST_ proviene de la extensión SQL/MM del estándar SQL (ISO/IEC 13249-3:2016 Part 3: Spatial), surgida en 1999, que tomó el modelo de Simple Features como base para especificar funcionalidades espaciales en SQL.

Simple Features

Implementaciones SQL

- Algunos SGBD's que lo implementaron:
 - **PostgreSQL:** PostGIS
 - **Mysql:** MySQL Spatial Extensions
 - **Oracle:** Oracle Spatial
 - **DB2:** IBM DB2 Spatial Extender
 - **SQL Server:** *built-in* desde SQL Server 2008

Introducción

Sistemas de coordenadas

- El **sistema de coordenadas** (*spatial reference system*) de una representación se define a través del SRID (Spatial Reference ID).
- El EPSG (European Petroleum Survey Group) estandariza los códigos de representación. Los más conocidos son:

SRID	Nombre	Comentarios
0	Ninguna	
4326	WGS84	Usado por el sistema de GPS.
7885	POSGAR 07	Cartografía oficial de la República Argentina.
4269	NAD83	El oficial de los Estados Unidos.
4258	ETRS89	Marco recomendado para la Unión Europea.

- Las áreas, longitudes y distancias de los objetos geométricos se calculan sobre el espacio bidimensional. PostGIS en particular permite castear las geometrías a **geografías**, cuyas medidas se calculan sobre el elipsoide definido en el sistema de referencia espacial, en $[m]$ ó $[m^2]$.

1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles
- GeoJSON
- KML y GML

4 Indexado de datos espaciales

5 Ejercicio en PostGIS

6 Bibliografía

- 1 Introducción
- 2 Simple Features
- 3 Otros formatos vectoriales
 - Shapefiles
 - GeoJSON
 - KML y GML
- 4 Indexado de datos espaciales
- 5 Ejercicio en PostGIS
- 6 Bibliografía

Otros estándares

Shapefiles

- El formato **ESRI shapefile** es un formato de archivo propietario creado por ESRI.
- ESRI es la compañía propietaria de ArcGIS, uno de los GIS más ampliamente utilizados.
- Es en realidad un formato multiarchivo, que divide al dataset geográfico en:
 - Un archivo **.shp** que contiene las formas geométricas asociadas a los objetos.
 - Un archivo **.shx** que almacena un índice espacial.
 - Un archivo **.dbf** de base de datos, que permite almacenar atributos de los objetos.
 - Un archivo **.prj** que almacena el sistema de coordenadas utilizado en formato WKT.
- PostGIS permite convertir un archivo shapefile a un listado de comandos SQL en base al estándar SQL/MM a través de la herramienta de línea de comando **shp2pgsql**.

1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles

- **GeoJSON**

- KML y GML

4 Indexado de datos espaciales

5 Ejercicio en PostGIS

6 Bibliografía

Otros estándares

GeoJSON

- **GeoJSON** (2008) es un estándar abierto basado en JSON (también RFC 7946 de la IETF).
- Sigue los conceptos del estándar Simple Features.
- Utiliza el sistema de coordenadas World Geodetic System 1984 (WGS84), y no permite otros.
- Fue motivado por el uso de datos espaciales en bibliotecas Javascript y en las bases de datos basadas en documentos (ej., MongoDB).
- En GeoJSON, una geometría es un documento JSON formado por dos campos: **type** y **coordinates**.

GeoJSON

Ejemplo

■ Ejemplo:

```
1 { "type": "Feature",  
2   "properties": { "población": 215000 },  
3   "geometry": {  
4     "type": "Polygon",  
5     "coordinates": [[  
6       [-109.05, 41.00],  
7       [-102.06, 40.99],  
8       [-109.04, 36.99],  
9       [-109.05, 41.00]  
10      ]]  
11   }  
12 }
```

1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles

- GeoJSON

- KML y GML

4 Indexado de datos espaciales

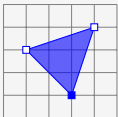
5 Ejercicio en PostGIS

6 Bibliografía

Otros estándares

KML y GML

- Ambos están basados en XML.
- Son de más alto nivel. Están pensados para representación 3D, y permiten especificar altitudes en forma absoluta ó con respecto a la superficie terrestre.
- **KML (Keyhole Markup Language)** es propiedad de Google y empleado en Google Earth.

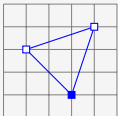


```
<Polygon id="ID">
  <outerBoundaryIs>
    <LinearRing>
      <coordinates>3.00, 1.00 1.00, 3.00 4.00, 4.00 3.00, 1.00
    </coordinates>
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
```

Otros estándares

KML y GML

- **GML (Geography Markup Language)** es del Open Geospatial Consortium (ISO 19136).



```
<gml:LinearRing>  
  <gml:coordinates>3.00, 1.00 1.00, 3.00 4.00, 4.00 3.00, 1.00  
  </gml:coordinates>  
</gml:LinearRing>
```

- GML permite especificar el sistema de coordenadas. En KML, en cambio, se utiliza exclusivamente WGS84.
- KML y GML son utilizados especialmente en *web services*.

GeoJSON, KML y GML

Conversiones

- El estándar SQL/MM ofrece funciones para procesar datos en formato GML:
 - ST_GeomFromGML
 - ST_AsGML
- Otras funciones típicas de conversión (aunque no estándar) son:
 - ST_GeomFromGeoJSON
 - ST_AsGeoJSON
 - ST_GeomFromKML
 - ST_AsKML

1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles
- GeoJSON
- KML y GML

4 Indexado de datos espaciales

5 Ejercicio en PostGIS

6 Bibliografía

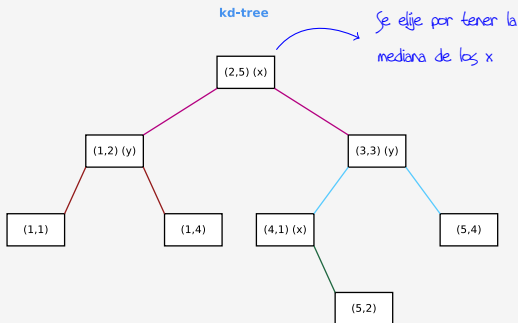
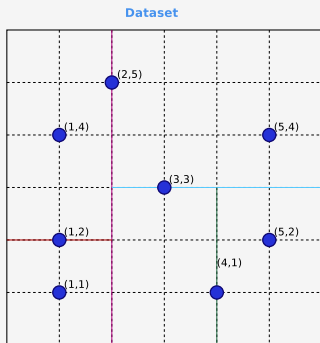
Indexado de datos espaciales

[SILB10 25.3.5; GM09 14.6; ELM16 26.3.4]

- Una de las funcionalidades más importantes de las bases de datos geoespaciales consiste en buscar y combinar objetos geográficos de manera eficiente.
- Se utilizan típicamente estructuras de búsqueda espaciales como:
 - *kd-trees*
 - *R-trees*
 - *Quad-trees*
 - Otras: R^* -trees, R^+ -trees, *X-trees*, ...
- Cada una de ellas presenta distintas ventajas y desventajas respecto a:
 - Los tipos de objetos geométricos que permiten almacenar.
 - Los tipos de consultas que permiten resolver.
 - La complejidad de las operaciones de inserción y eliminación.

kd-trees

- Los **kd-trees** son árboles binarios multidimensionales que indexan **puntos** de un espacio \mathbb{R}^k .



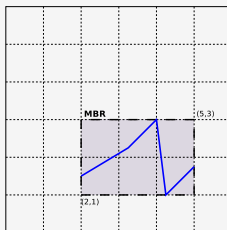
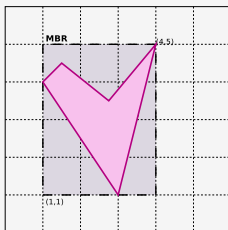
kd-trees

- Cada nodo no-hoja está asociado a un punto del set de datos $\bar{x} = (x_1, \dots, x_k)$ y a una dimensión $1 \leq j \leq k$.
 - Divide al espacio en dos partes utilizando un hiperplano perpendicular al eje de coordenadas x_j que pasa por el punto (ecuación $x_j = \bar{x}_j$).
 - Cada uno de sus dos hijos indexará una de estas dos partes.
- Los nodos hoja están asociados directamente a un punto del set de datos.
- Los *kd-trees* permiten típicamente hacer:
 - Búsquedas por rango
 - Qué puntos del dataset están dentro de una región dada
 - $\{\bar{x}^{inside}\} = T.range_search(\bar{x}^{min}, \bar{x}^{max})$
 - Búsquedas de vecino más cercano
 - Cuál es el punto del dataset más cercano a un punto del espacio
 - $\bar{x}^{nearest} = T.nearest_neighbor(\bar{x})$

R-trees

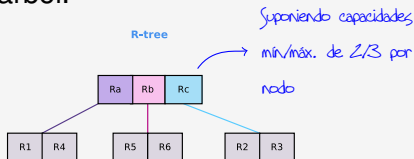
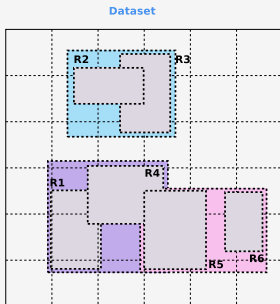
Minimum bounding rectangles

- Los *R-trees* indexan objetos geométricos (puntos, líneas, polígonos, etc.) a través de su *minimum bounding rectangle (MBR)*.
- El *minimum bounding rectangle* de un objeto k -dimensional es el *ortopedro* k -dimensional minimal que lo contiene (en 2 dimensiones es un rectángulo).
- Dicho ortopedro puede representarse a partir de un par de puntos (\bar{x}^{min} , \bar{x}^{max})



R-trees

- Las hojas del *R-tree* están asociadas a MBRs de objetos geométricos del set de datos. Los nodos no-hoja están asociados a MBRs que contienen a todos los MBRs de sus hijos.
- Los MBRs no representan particiones del espacio. Puede haber solapamiento entre MBRs de distintas entradas del árbol.
- Al igual que los *B-trees*, los *R-trees* son balanceados y agrupan varias entradas en cada nodo del árbol.



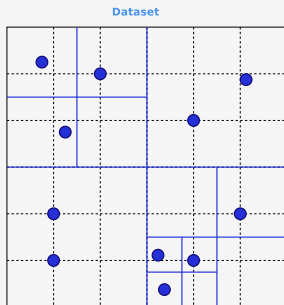
R-trees

Consultas

- Los *R-trees* permiten resolver:
 - Búsquedas de pertenencia
 - Qué polígono/s P del dataset contienen un punto dado
 - $\{P^{containing}\} = T.contains(\bar{x})$
 - Búsquedas de intersecciones
 - Qué polígonos P del dataset se intersecan con un polígono dado
 - $\{P^{intersecting}\} = T.intersects(P)$
- Al ser capaces de manejar todo tipo de objetos geométricos y poder resolver consultas típicamente vinculadas con **juntas espaciales**, los *R-trees* se han vuelto los más populares en los SGBD's que soportan datos espaciales.

Quad-trees

- En los *Quad-trees* el espacio se descompone en cuadrantes disjuntos.
- Cuando un cuadrante llega al máximo de su capacidad en términos de cantidad de objetos, el mismo vuelve a particionarse en cuatro.
- Se utilizan principalmente para indexar puntos geométricos o datos *raster* en dos dimensiones.



Suponiendo que la
capacidad máxima del
cuadrante es 2

1 Introducción

2 Simple Features

3 Otros formatos vectoriales

- Shapefiles
- GeoJSON
- KML y GML

4 Indexado de datos espaciales

5 Ejercicio en PostGIS

6 Bibliografía

Ejercicio en PostGIS

- 1 Es necesario tener instalada la extensión PostGIS para PostgreSQL (<https://postgis.net/install/>).
- 2 Y luego habilitarla dentro de la base de datos (*database*) de Postgres que usaremos (sólo la primera vez):

```
CREATE EXTENSION postgis;
```

- 3 Creación de una tabla con atributo espacial:

```
CREATE TABLE lagos_argentina (nombre_lago varchar,  
                                poligono geometry);
```

- 4 Inserción en una tabla:

```
INSERT INTO lagos_argentina VALUES('Lago_Lacar',  
                                     ST_ASBinary('POLYGON(...)'));
```

- 5 Creación de un índice R-Tree:

```
CREATE INDEX nombre_indice  
ON tabla USING GIST (atributo_geometrico);
```


Ejercicio en PostGIS

■ Trabajaremos con dos fuentes de datos:

- 1 La base de datos de **Departamentos de la República Argentina** ofrecida por el Instituto Geográfico Nacional (IGN), en formato **shapefile**.
 - <http://www.ign.gob.ar/NuestrasActividades/InformacionGeoespacial/CapasSIG>
- 2 Datos de la **Red Federal de Fibra Óptica** ofrecidos por ARSAT en formato **JSON**.
 - <http://datos.arsat.com.ar/visualizations/28598/fibra-optica-en-servicio/>
- 3 Ambos datasets se encuentran subidos al Campus.

Ejercicio en PostGIS

Ejercicio

- 1 Convierta los datos sobre Departamentos de la República Argentina en un archivo de sentencias SQL (**creacion_departamentos.sql**) que cree una tabla de **Departamentos**, utilizando el comando **shp2pgsql**. Tenga en cuenta configurar el SRID de las geometrías. Observe el tamaño del archivo, la estructura de tabla a crear y el formato en que se carga cada polígono en la base de datos.
- 2 Luego ejecute las sentencias del archivo con el comando **psql** ó bien desde el **pgadmin**.

Nota: Hoja de metadatos provista por el IGN en:

<http://www.ign.gob.ar/descargas/geodatos/pais.pdf>.

Ejercicio en PostGIS

Ejercicio

- 3** Escriba las siguientes consultas en lenguaje SQL:
 - (a) Contar la cantidad de departamentos existentes en el país.
 - (b) Listar para los 10 departamentos de menor superficie del país, su identificador (gid), su nombre (fna) y su superficie.

Ejercicio

Ejercicio

- 4 Cada departamento tiene su propio territorio, y por lo tanto dos departamentos no tienen superficie común entre sí (a lo sumo pueden ser linderos). Para tener una idea de los errores presentes en nuestra representación con polígonos, liste aquellos pares de departamentos que comparten superficie entre sí. Extraiga una conclusión.
- 5 Habrá observado que la consulta anterior tiene un costo relativamente alto. Construya un índice R-Tree sobre el atributo **geom** de la tabla anterior, y vuelva a ejecutar la consulta anterior. ¿Mejoró el tiempo de ejecución?

Ejercicio en PostGIS

- Los datos de la red de fibra óptica no tienen formato geográfico, sino que vienen en un JSON array en que cada elemento representa a un tramo e incluye tres strings:
 - Un nombre de tramo
 - Un comentario opcional
 - Un listado de coordenadas en el mismo formato utilizado por `<gml:coordinates>` en GML:

```
1  [
2    [ "La Quiaca — Tres Cruces", "", "x0,y0,z0 x1,y1,z1 ..." ],
3    [ "Tres Cruces — Humahuaca", "GEOTIS", "x0,y0,z0 x1,y1,z1 ..." ],
4    ...
5  ]
```

Ejercicio en PostGIS

- Cargaremos estos datos en una tabla de PostgreSQL previa conversión al formato GML, y luego utilizando la función **ST_GeomFromGML()**.

```
CREATE TEMPORARY TABLE temp (filas TEXT);

COPY temp FROM 'red.json'; #Poner path absoluto

CREATE TABLE Red AS
SELECT elemento->>0 AS nombre_tramo,
       elemento->>1 AS comentario,
       ST_GeomFromGML('<gml:LineString><gml:coordinates>' ||
                     (elemento->>2) ||
                     '</gml:coordinates></gml:LineString>', 4326)
       AS recorrido

FROM (
  SELECT json_array_elements(filas::json) AS elemento FROM
    temp
) filas;
```

Ejercicio en PostGIS

Ejercicio

- 6** Para los 10 tramos más largos de la red de fibra, indique su nombre y longitud.
- 7** Combinando la información de esta tabla con la de los departamentos del país, encuentre los 20 departamentos que contienen mayor tendido de fibra óptica relativo a su superficie, y para cada uno de ellos indique su nombre y la longitud de fibra óptica total tendida en relación con su superficie.

- 1 Introducción
- 2 Simple Features
- 3 Otros formatos vectoriales
 - Shapefiles
 - GeoJSON
 - KML y GML
- 4 Indexado de datos espaciales
- 5 Ejercicio en PostGIS
- 6 Bibliografía

Bibliografía

Estándar Simple Features

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

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

Open Geospatial Consortium, 2011.

Documentación de PostGIS

https://postgis.net/docs/using_postgis_dbmanagement.html

PostgreSQL, 2018.

Capítulo 4: Using PostGIS: Data Management and Queries.

Bibliografía

[ELM16] Fundamentals of Database Systems, 7th Edition.

R. Elmasri, S. Navathe, 2016.

Capítulo 26.3

[GM09] Database Systems, The Complete Book, 2nd Edition.

H. García-Molina, J. Ullman, J. Widom, 2009.

Capítulo 14.6

[SILB10] Database System Concepts, 6th Edition.

A. Silberschatz, H. Korth, S. Sudarshan, 2010.

Capítulo 25.3