ANÁLISIS ESPACIAL CON POSTGIS

Práctica 3

Bases de Datos Distribuidas

Rodríguez García Alan Julian

Moreno Tagle Iván Raphael

Introducción

A continuación se describen los resultados obtenidos de seguir el manual de la *Práctica 3* de la materia de **Bases de Datos Espaciales**. En esta ocasión, se describe el resultado de un total de 13 consultas espaciales, con diagramas generados mediante la herramienta OPENJUMP. No se describe el contenido de la carga de datos ni el proceso de crear el DDL del modelo, ya que consideramos de por si amplio el contenido del reporte.

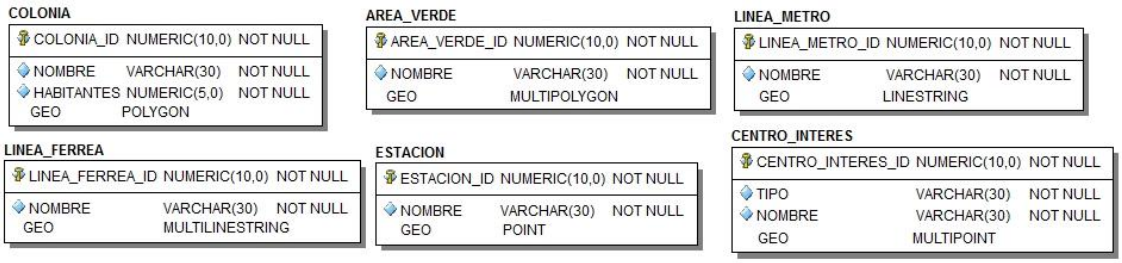
Objetivo

El objetivo de esta práctica es realizar análisis espacial de un área geográfica haciendo uso del estándar OpenGIS creado para representar, almacenar, y explotar información espacial a través del uso de la extensión espacial de PostgreSQL: PostGIS.

El análisis se basa en la construcción de una serie de sentencias SQL que permiten familiarizarse con la jerarquía de ADTs así como del conjunto de funciones disponibles que permiten llevar a cabo esta tarea.

Desarrollo

C0: Modelo Relacional



Modelo entidad relación que define las reglas de negocio al cual se le realizaron las consultas espaciales.

Consulta 1:

Se desea conocer para cada colonia, el número de estaciones que pertenecen a la red del metro que pasan por cada colonia. Generar una consulta que muestre el nombre de la colonia (no considerar las estaciones que son exclusivas de las líneas férreas), y el número de estaciones. Ordenar por el nombre de la colonia. En caso que la estación esté en los límites de la colonia, está también se deberá incluir. Emplear sintaxis estándar.

```
espaciales=> select count(*), c.nombre as colonia
espaciales-> from colonia c join estacion e
espaciales-> on st_covers(c.geo, e.geo)
espaciales-> join (select st_union(geo) as geo_u from linea_metro) lm
espaciales-> on st_covers(lm.geo_u, e.geo)
espaciales-> group by c.nombre;
 count | colonia
-------+----------
     2 | AGUIRRE
     4 | ALLENDE
     3 | CARRANZA
     7 | CENTRO
     2 | HIDALGO
     1 | ITURBIDE
     3 | JUAREZ
     1 | MONTES
     2 | MORELOS
     3 | SUAREZ
     2 | VILLA
     4 | ZAPATA
     2 | ZARAGOZA
(13 rows)

espaciales=>
```

Consulta 2:

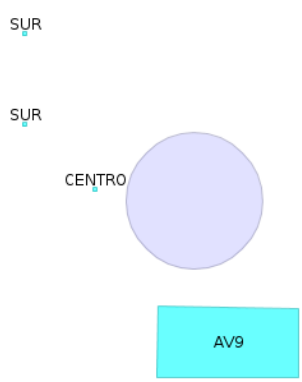
Debido a una amenaza de bomba en la estación del metro AM1, se ha decidido acordonar un área con radio de 1 km a la redonda. Generar una consulta que muestre todos nombres de los centros de interés, su tipo, así como los nombres de áreas verdes que pueden verse afectados que estén a una distancia menor o igual a 3 km del área acordonada. Adicional a estos 2 campos, mostrar la distancia que existe entre el área acordonada y el objeto. Emplear sintaxis estándar. Recomendaciones: Emplear la función DWithin. El reporte deberá verse así:

Nombre	Tipo	Distancia
AVX	area verde	X
SUR	Hospital	X

```
espaciales=> with q1 as(
espaciales(> select st_buffer(e.geo, 1) as geo from estacion e
espaciales(> where e.nombre='AM1'
espaciales(>      )
espaciales-> select ci.nombre, ci.tipo as TIPO, st_distance((ci.dump).geom, st_boundary(q1.geo)) as Distancia
espaciales-> from
espaciales-> (
espaciales(> select st_dump(geo) as dump , nombre, tipo from centro_interes
espaciales(> )ci
espaciales-> join q1 on st_dwithin((ci.dump).geom, st_boundary(q1.geo), 3)
espaciales-> UNION
espaciales-> select av.nombre, 'area_verde' as tipo, st_distance(av.geo, st_boundary(q1.geo)) as Distancia
espaciales-> from area_verde av
espaciales-> join q1 on st_dwithin(av.geo, st_boundary(q1.geo), 3);
 nombre |     tipo      |      distancia
--------+---------------+--------------------
 AV9    | area_verde    | 0.554797040048294
 CENTRO | UNIVERSIDAD   | 0.473274771213633
 SUR    | HOSPITAL      |   1.7359044722064
 SUR    | HOSPITAL      |  2.49928853008146
(4 rows)
```

Consulta 3:

Generar un diagrama con OpenJump que muestre la geometría del área acordonada del ejercicio anterior, incluir los tópicos de áreas verdes y centros de interés para verificar de forma gráfica los resultados del ejercicio anterior



Consulta 4:

A todas las colonias que tengan 3 o más edificios que pertenezcan a la red de unidades habitacionales, se les deberá otorgar un mayor presupuesto para realizar tareas de mantenimiento. Determinar los nombres de las colonias y el número de edificios que contiene a las que se les dará un mayor presupuesto. Los edificios que estén justo en los límites de las colonias no se consideran en el conteo. Emplear sintaxis estándar. Recomendación: Emplear ST_DUMP. Observar la colonia Carranza, la consulta debe obtener 3 edificios.

```
espaciales=> select q1.nombre, count(*)
espaciales-> from
espaciales-> (
espaciales(> select st_difference(c.geo, av1.geo_n) as diff, c.nombre
espaciales(> from colonia c,
espaciales(>
espaciales(> (
espaciales(> select st_union(av.geo) as geo_n from area_verde av
espaciales(>

espaciales(> ) av1
espaciales(>
espaciales(> ) q1 join
espaciales-> (
espaciales(> select st_dump(geo) as dump
espaciales(> from centro_interes ci
espaciales(> where ci.tipo = 'UNIDAD HABITACIONAL'
espaciales(> )q2 on st_contains(q1.diff, (q2.dump).geom)
espaciales-> group by q1.nombre
espaciales-> HAVING count(*)>= 3;
  nombre   | count
----------+-------
 SUAREZ   |     4
 CARRANZA |     3
 VILLA    |     3
(3 rows)
```

Consulta 5:

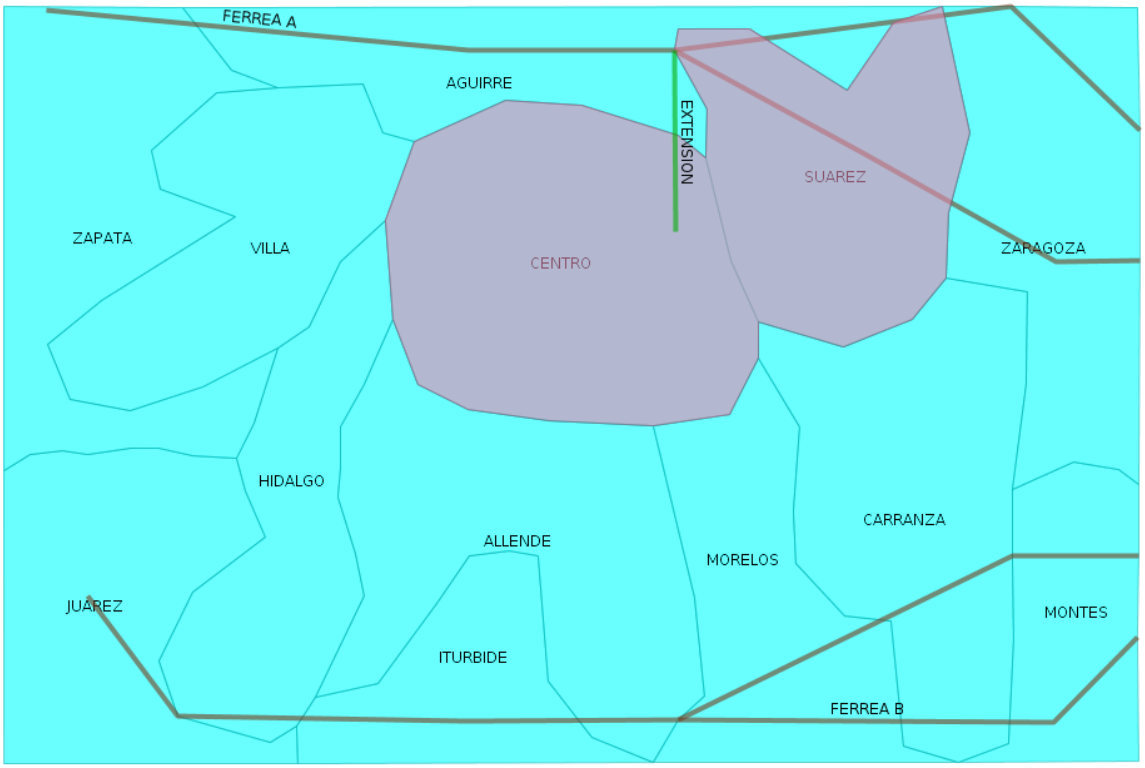
Algunas estaciones de la red del metro y de la red férrea están ubicadas en alguna área verde. Determinar los nombres de las estaciones, el nombre del área verde y el nombre de la línea férrea o del metro a la que pertenecen, así como su ubicación (geometría de la estación). Emplear sintaxis estándar. Ordenar por el nombre de la línea. Ejemplo:

Nombre línea	Estación	Área verde	Ubicación
AZUL	A3	av8	POINT(x y)
FERREA B	FB1	av6	POINT(x y)

```
espaciales=>
espaciales=> select *
espaciales-> from
espaciales->   (select e.nombre as ESTACION, lm.nombre as Nombre_linea, av.nombre as Area_Verde, st_astext(e.geo) as UBICACION
espaciales(>   from estacion e join linea_metro lm on st_covers(lm.geo, e.geo)
espaciales(>   join area_verde av on st_covers(av.geo, e.geo)
espaciales(>   UNION
espaciales(>   select e.nombre as ESTACION, lf.nombre as Nombre_linea, av.nombre as Area_Verde, st_astext(e.geo) as UBICACION
espaciales(>   from estacion e join linea_ferrea lf on st_covers(lf.geo, e.geo)
espaciales(>   join area_verde av on st_covers(av.geo, e.geo)) q1
espaciales-> order by nombre_linea;
 estacion | nombre_linea | area_verde |        ubicacion
----------+--------------+------------+-------------------------
 A3       | AZUL         | AV8        | POINT(9.00705 6.95887)
 FFB1     | FERREA B     | AV6        | POINT(11 1)
 R1       | ROJA         | AV1        | POINT(3.02349 14.9965)
(3 rows)
```

Consulta 6:

Suponga que las colonias Centro y Suarez se van a fusionar. Por otro lado, se planea extender la línea férrea A agregando un nuevo segmento de recta que va del punto donde se bifurca la línea hacia el centro geométrico de la nueva colonia (fusionada). Generar una consulta SQL para construir un diagrama en OpenJump que dibuje el nuevo segmento de recta. Agregar al diagrama el tópico de las líneas férreas para observar que efectivamente la línea férrea se extenderá.



Consulta 7:

Un turista se encuentra en la colonia Centro. El turista desea hospedarse en algún hotel que esté ubicado en alguna colonia vecina. Determine los nombres de las colonias, los nombres de los hoteles y sus ubicaciones donde el turista puede hospedarse. Emplear sintaxis estándar.

```
select cv.nombre, ci.nombre, st_astext(ci.geo) as UBICACION
from colonia c
join colonia cv on st_touches(c.geo, cv.geo)
join centro_interes ci
on st_covers(cv.geo,ci.geo)
where ci.tipo = 'HOTEL'
and c.nombre = 'CENTRO';
```

	nombre character varying(30)	nombre character varying(30)	ubicacion text
1	HIDALGO	SOL-B	MULTIPOINT(5.84 2.96)
2	MORELOS	SOL-B	MULTIPOINT(18.0141 4.03995)
3	VILLA	SOL-A	MULTIPOINT(7.99217 14.9965)

Consulta 8:

Generar una consulta que muestre el nombre de la línea del metro con el mayor número de estaciones. Incluir en el reporte el número de estaciones con las que cuenta.

```
espaciales=>
espaciales=> select lm.nombre, count(*) as num_estaciones
espaciales-> from linea_metro lm
espaciales-> join estacion e
espaciales-> on st_covers(lm.geo,e.geo)
espaciales-> group by lm.nombre
espaciales-> having count(*) = (
espaciales(>                    select max(cuenta)
espaciales(>                    from (
espaciales(>                      select count(*) as cuenta
espaciales(>                      from linea_metro lm1
espaciales(>                      join estacion e1
espaciales(>                      on st_covers(lm1.geo,e1.geo)
espaciales(>                      group by lm1.nombre
espaciales(>                    ) q1
espaciales(>                );
 nombre | num_estaciones
--------+----------------
 AZUL   |               9
 CAFE   |               9
(2 rows)

espaciales=>
```

Consulta 9:

Se desea cercar con una malla metálica a las geometrías que forman al área verde AV4. Para cada una de sus geometrías calcular en KM, la longitud de malla que se requiere. El reporte debe contener las siguientes columnas:

Núm. Geometría	longitud total
1	xx
2	xx

Los números 1,2 y 3 corresponden con en número asignado a la geometría. Recomendación, emplear ST_DUMP y el arreglo path para extraer el índice de la geometría.

```
espaciales=> select (q1.dump).path[1] as num_geometria,
espaciales-> st_length(st_boundary((q1.dump).geom)) as longitud_total
espaciales-> from
espaciales->   (select st_dump(av.geo) as dump, av.nombre
espaciales(>   from area_verde av
espaciales(>   where av.nombre = 'AV4')q1;
 num_geometria |  longitud_total
---------------+------------------
             1 | 9.75009902893913
             2 |          3.89114
             3 | 3.99803945565225
(3 rows)

espaciales=>
```

Consulta 10:

Se han marcado 2 áreas imaginarias donde la delincuencia ha aumentado, definidas por los rectángulos R1 y R2 que forman las diagonales d1: P1(6,6), P2 (14,14) y d2: P1 (15,15), P2 (27,27). Generar una sentencia que muestre los nombres de los centros de interés completamente contenidos en cualquiera de los rectángulos. El reporte debe ser similar al siguiente:

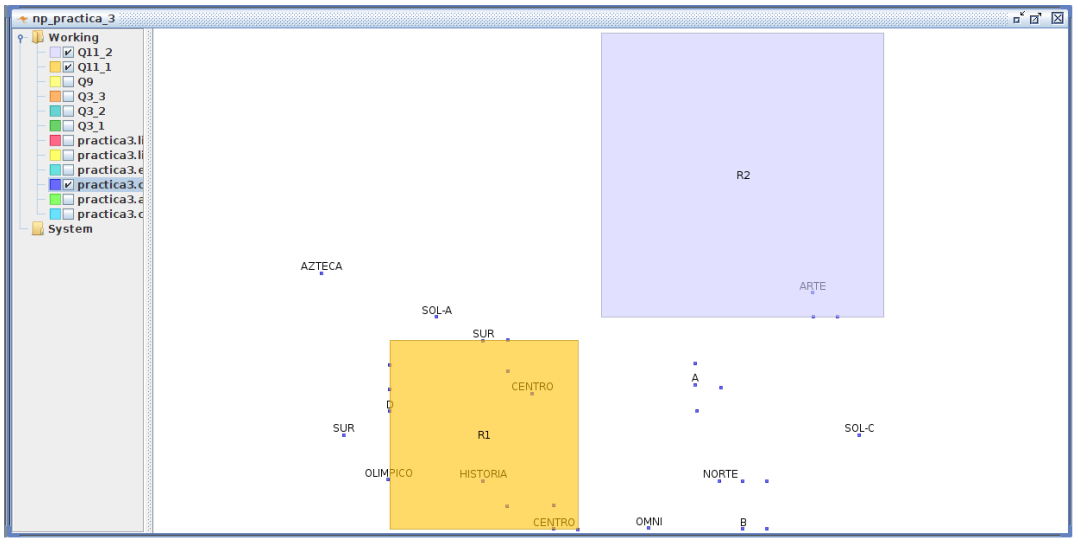
Nombre	Tipo	Nombre rectángulo
OLIMPICO	ESTADIO	R1

R1 es el nombre del rectángulo imaginario. Nota: no se requiere crear tablas adicionales para crear los rectángulos.

```
espaciales=> with q1 as(
espaciales(>  select 'R1'as nombre,
espaciales(>      st_envelope(st_makebox2d(
espaciales(>          'POINT(6 6)'::geometry,
espaciales(>          'POINT(14 14)'::geometry
espaciales(>         )
espaciales(>      ) as geo
espaciales(>  union
espaciales(>  select 'R2'as nombre,
espaciales(>      st_envelope(st_makebox2d(
espaciales(>          'POINT(15 15)'::geometry,
espaciales(>          'POINT(27 27)'::geometry
espaciales(>         )
espaciales(>      ) as geo
espaciales(> )
espaciales-> select ci.nombre, ci.tipo, q1.nombre
espaciales-> from  q1 join centro_interes ci
espaciales-> on st_contains(q1.geo,ci.geo)
espaciales-> where q1.nombre = 'R1'
espaciales-> union
espaciales-> select ci.nombre, ci.tipo, q1.nombre
espaciales-> from  q1 join centro_interes ci
espaciales-> on st_contains(q1.geo,ci.geo)
espaciales-> where q1.nombre = 'R2';
  nombre  |     tipo     | nombre
----------+--------------+--------
 CENTRO   | UNIVERSIDAD  | R1
 HISTORIA | MUSEO        | R1
(2 rows)
```

Consulta 11:

Generar un diagrama en OpenJump que muestre los 2 rectángulos del ejercicio anterior, agregar el tópico de centros de interés para comprobar los resultados obtenidos anteriormente.



Consulta 12:

Determinar el nombre y la ubicación de las estaciones de la red del metro en donde un pasajero puede cambiar de línea. El reporte debe mostrarse así:

Nombre Estación	Ubicación	Línea Origen	Línea Destino
AM1	POINT(13.5106 11.5699)	AZUL	MORADA
AM1	POINT(13.5106 11.5699)	MORADA	AZUL

Observar que una estación puede aparecer 2 veces debido al sentido de recorrido de la linea. Ordenar los resultados por el nombre de la estación, no emplear WITH ni subqueries, emplear sintaxis estándar.

```
espaciales=> select e.nombre, st_asText(e.geo) as UBICACION, lm.nombre as LINEA_ORIGEN, lm1.nombre as LINEA_DESTINO
espaciales-> from linea_metro lm
espaciales-> join linea_metro lm1
espaciales-> on st_crosses(lm.geo,lm1.geo)
espaciales-> join estacion e
espaciales-> on st_covers(lm.geo,e.geo)
espaciales-> and st_covers(lm1.geo,e.geo)
espaciales-> order by e.nombre;
 nombre |        ubicacion         | linea_origen | linea_destino
--------+--------------------------+--------------+---------------
 AM1    | POINT(13.5106 11.5699)   | AZUL         | MORADA
 AM1    | POINT(13.5106 11.5699)   | MORADA       | AZUL
 CA1    | POINT(14.5677 13.0505)   | CAFE         | AZUL
 CA1    | POINT(14.5677 13.0505)   | AZUL         | CAFE
 CM1    | POINT(12.0094 14.0024)   | CAFE         | MORADA
 CM1    | POINT(12.0094 14.0024)   | MORADA       | CAFE
 CR1    | POINT(8.01331 11.9295)   | ROJA         | CAFE
 CR1    | POINT(8.01331 11.9295)   | CAFE         | ROJA
 RA1    | POINT(11.9248 9.98355)   | AZUL         | ROJA
 RA1    | POINT(11.9248 9.98355)   | ROJA         | AZUL
(10 rows)
```

Consulta 13:

Existen algunas áreas verdes que están completamente contenidas en las colonias. Los límites de algunas de estas áreas tocan en uno o más puntos a los límites de las colonias. Empleando la función `st_relate`, determine el nombre de las colonias, el nombre de las áreas verdes y el área de las áreas verdes que cumplen con las características mencionadas. No emplear los valores *,T.

```
select c.nombre, av.nombre, st_area(av.geo)
from colonia c
join area_verde av
on st_relate(av.geo,c.geo,'2FF10F212')
or st_relate(av.geo,c.geo,'2FF11F212')
or st_relate(av.geo,c.geo,'2FF1FF212');
```

	nombre character varying(30)	nombre character varying(30)	st_area double precision
1	ZAPATA	AV1	5.968684626
2	AGUIRRE	AV2	4.480455535
3	SUAREZ	AV3	2.87927056
4	CARRANZA	AV4	7.58947501750001
5	ALLENDE	AV5	3.972592651
6	ITURBIDE	AV6	3.043337739975
7	JUAREZ	AV7	1.9974920301
8	CENTRO	AV9	2.0932757125

Conclusiones

Definitivamente se cumplió el objetivo, pues se logró obtener correctamente el resultado de las trece consultas espaciales, que aunque significaron un reto, sin duda nos ayudaron a poder comprender la mayoría de los subtemas del tema 3 de la materia de Bases de Datos Espaciales, pues no hay mejor forma de aprender que practicando.

Bibliografía

-Manual Práctica 3 de **Bases de Datos Espaciales**. Ing. Jorge A. Rodríguez Campos

-Documentación POSTGIS: http://postgis.refractions.net/documentation/manual-1.4/ch04.html#spatial_ref_sys