

**PRÁCTICA 3.**  
**ANÁLISIS ESPACIAL CON POSTGIS**

La práctica se entrega en equipos máximo 3 personas.

**1.1. 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.

**1.2. DESARROLLO DE LA PRÁCTICA.**

**1.2.1. Definición del área geográfica.**




La figura al final del documento muestra un mapa con un conjunto de tópicos de interés para una ciudad organizados en las siguientes capas:

- Colonias: Representadas por polígonos.
- Áreas verdes: representadas por polígonos V1, etc. Observar que algunas áreas verdes están formadas por más de un polígono.
- Líneas del metro: Representadas por las líneas de colores mostradas en la figura
- Líneas férreas: Observar que algunas de ellas presentan bifurcaciones.
- Estaciones. Representadas por puntos, en donde el tren o el metro realizan paradas.
- Centros de interés. Representados por un conjunto de puntos. Un centro de interés es un lugar donde pueden existir actividades recreativas, así como lugares para vivir. Los centros de interés se clasifican por tipo:
  - Museo
  - Hospital
  - Estadio
  - Unidad habitacional
  - Hotel
  - Campamento militar
  - Universidad
 Observar que algunos de estos centros son representados por más de un punto.



**1.2.2. Diseño de la base de datos espacial.**

La información de estos tópicos se almacena empleando las siguientes estructuras:



**COLONIA**

	COLONIA_ID	numeric(10,0)	NOT NULL
	NOMBRE	varchar(30)	NOT NULL
	HABITANTES	numeric(5,0)	NOT NULL



**AREA\_VERDE**

	AREA_VERDE_ID	numeric(10,0)	NOT NULL
	NOMBRE	varchar(30)	NOT NULL



**LINEA\_METRO**

	LINEA_METRO_ID	numeric(10,0)	NOT NULL
	NOMBRE	varchar(30)	NOT NULL




**LINEA\_FERREA**

	LINEA_FERREA_ID	numeric(10,0)	NOT NULL
	NOMBRE	varchar(30)	NOT NULL

**ESTACION**

	ESTACION_ID	numeric(10,0)	NOT NULL
	NOMBRE	varchar(30)	NOT NULL

**CENTRO\_INTERES**

	CENTRO_INTERES_ID	numeric(10,0)	NOT NULL
	TIPO	varchar(30)	NOT NULL
	NOMBRE	varchar(30)	NOT NULL

**Actividad 1.**

- A. Para crear estas estructuras, se hará uso de la base de datos “espaciales” creada en la práctica anterior. Las tablas de esta práctica deberán crearse en un nuevo esquema llamado “practica3”. Para ello, desde pgAdmin estando conectado a la base de datos “espaciales”, abrir un nuevo editor SQL, crear un nuevo esquema empleando las siguientes instrucciones:

```
drop schema if exists practica3 cascade;
create schema practica3;
```

- B. Para crear las tablas en el esquema `practica3`, existen varias estrategias. La más sencilla es modificar una variable de sesión para evitar anteponer el prefijo “`practica3`” en cada sentencia SQL. Por lo anterior, antes de realizar cualquier operación SQL en el editor, se deberá ejecutar la siguiente instrucción:

```
set search_path to practica3, public;
```

Con esta instrucción, todas las sentencias SQL se realizarán considerando el esquema `practica3`.

Importante: No **olvidar ejecutar esta instrucción cada vez que se abre un nuevo editor SQL, ya que su valor se pierde al cerrar pgAdmin o al iniciar un nuevo editor.**

- C. Generar el código DDL y crear las tablas con base a la figura anterior. (No es necesario incluir el código SQL en el reporte).  
 D. Generar una sentencia SQL que agregue un nuevo campo “geo” para cada tabla empleando la función `addGeometryColumn`

```
AddGeometryColumn (
  <schema_name>,
  <table_name>,
  <column_name>,
  <srid>,
  <type>,
  <dimension>
)
```

- `Schema_name`: en caso de que no se especifica, el valor del esquema se asigna a “public”.
- `Table_name`: Nombre de la tabla a la que se le agregará el ADT.
- `Column_name`: Nombre de la columna que representa al ADT.
- `SRID`: Identificador del sistema de referencia asociado al ADT. Su valor corresponde con algún sistema registrado en la tabla `SPATIAL_REF_SYS`. Para efectos del curso, le asignaremos 0 (sistema de referencia indefinido).
- `Type`: el tipo de ADT, corresponde con los nombres de los ADTs empleados para realizar su representación en formato WKT
- `Dimensión`: La dimensión del sistema de referencia. Para efectos del curso es 2, es decir, un sistema de coordenadas X,Y (no confundir con la dimensión del objeto espacial).

**Ejemplo:**

```
select addgeometrycolumn('practica3','estacion','geo',0,'POINT',2);
```

**C1: Incluir en el reporte** las sentencias SQL que agregan la columna geo a cada tabla.

- E. Para garantizar que una geometría es válida, en la práctica anterior se revisó la función `st_isvalid`. Sin embargo, es posible agregar una restricción a cada tabla para que la verificación se realice en automático al momento de insertar un nuevo registro. Para cada tabla ejecutar la siguiente instrucción:

```
alter table <nombre_tabla> add constraint geometry_valid_check check
(st_isvalid(<nombre_campo>));
```

- `nombre_tabla`: Nombre de la tabla a la que se agregará el constraint para validar la geometría.
- `nombre_campo`: Nombre de la columna (ADT) a la que se le aplica la restricción.

**Ejemplo:**

```
alter table estacion add constraint geometry_valid_check check (st_isvalid(geo));
```

**C2: incluir en el reporte** las sentencias SQL que agregan estos constraints.

### 1.2.3. Mapeo de objetos espaciales.

#### Actividad 2.

Se proporciona para el poblado de la base de datos un archivo con el mapeo de puntos para cada objeto. A partir de este conjunto de puntos, se deberán realizar las siguientes actividades:

- Para cada objeto generar una cadena en formato WKT
- Insertar los datos en su correspondiente tabla.
- **C3: Incluir en el reporte** 5 sentencias SQL de objetos de diferente tipo.

#### 1.2.4. Instalación de OpenJump

Open Jump es un sistema de información geográfica que permite realizar la interacción entre una base de datos espacial entre muchas otras funcionalidades que permite generar mapas y análisis espacial de forma gráfica. Seguir los siguientes puntos para realizar su instalación:

1. Obtener el software de la siguiente página en su versión más reciente: <http://www.openjump.org/> En la página de descargas, obtener el archivo jar OpenJUMP-Installer-**X.X.X-rxxxx**-PLUS.jar , donde **X.X.X-rxxxx** corresponde con los valores de la versión más reciente.
2. OpenJump requiere la instalación del JDK (Java Development Kit). Para ello, en caso de no contar con una versión de Java 6 o mayor, realizar los siguientes puntos:
3. Descargar la versión más reciente del JDK en <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
4. Descomprimir el archivo en cualquier directorio.
5. Copiar la carpeta generada en alguna ruta destino del sistema, se recomienda en /opt:  

```
sudo mv jdk1.X.X_XX /opt
```

X.X\_XX se refiere a la versión .
6. Agregar y/o modificar las variables de entorno en el archivo:
  - a. /etc/bash.bashrc Para Mint/Ubuntu
  - b. /etc/profile para Fedora.

#### Ejemplo:

```
sudo nano /etc/bash.bashrc
```

#### #Variables de entorno para java

```
export JAVA_HOME=/opt/jdk1X.X.X_ X
```

Sustituir X.X.X por el valor de la versión escargada.

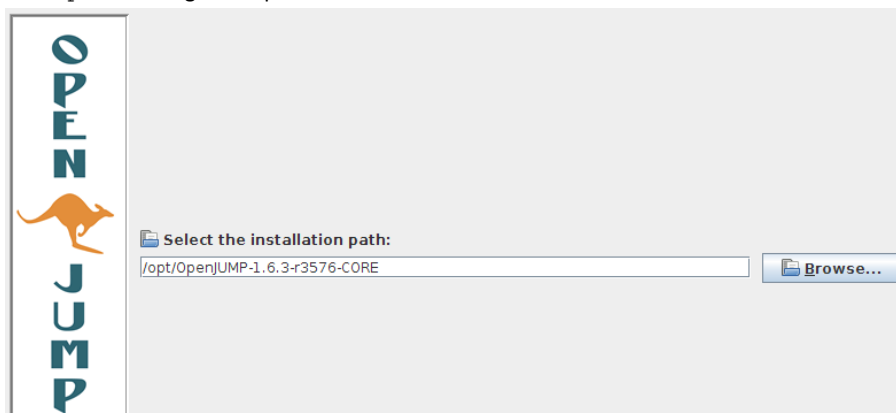
#### #agrega el directorio bin del jdk a la variable PATH

```
export PATH=$JAVA_HOME/bin:$PATH
```

7. Para actualizar los cambios en la terminal actual, ejecutar `source /etc/bash.bashrc` o `source /etc/profile` según corresponda. En su defecto cerrar terminales y abrir una nueva.
8. Para iniciar el instalador de OpenJump, ejecutar las siguientes instrucciones, en el directorio donde se descargó el archivo jar. Se ejecuta como root para instalar OpenJump en /opt  

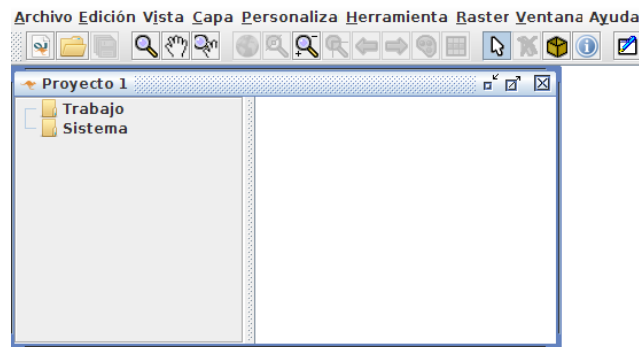
```
sudo su
```

```
java -jar OpenJUMP-Installer-X.X.X-rxxxx-CORE.jar
```
9. Aparecerá una serie de pantallas para realizar su instalación (seleccionar el idioma inglés). En la pantalla de selección del directorio de instalación se recomienda instalarlo en /opt. Ver la siguiente pantalla:



10. En Linux no funcionan los accesos directos que se generan. Se pueden omitir.
11. Para ejecutar OpenJump, ejecutar la siguiente instrucción, asumiendo que la ruta de instalación es la mostrada en la figura. No es necesario ejecutar como root.  

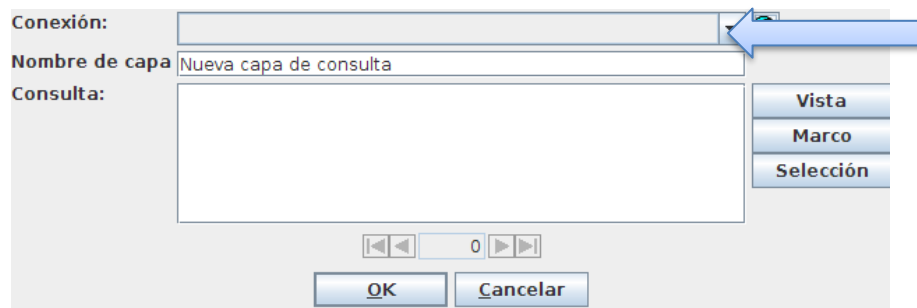
```
sh /opt/OpenJUMP-X.X.X-r3576-CORE/bin/oj_linux.sh &
```
12. Aparecerá la pantalla principal de OpenJump.



### 1.2.5. Conexión a bases de datos espaciales con OpenJump.

Una de las funcionalidades de OpenJump es la posibilidad de ejecutar consultas SQL desde la herramienta cuyo resultado sea una geometría. OpenJump puede mostrar el resultado de forma gráfica. Para realizar la conexión a la base de datos realizar las siguientes acciones:

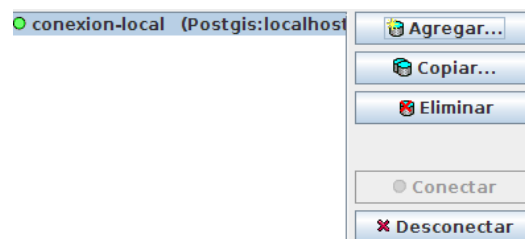
1. Seleccionar Archivo -> Ejecutar consulta de almacén de datos. Aparecerá una ventana similar a la siguiente. Hacer clic en el ícono de conexión a base de datos.



2. Hacer clic en el botón Agregar, y proporcionar los datos de la conexión como se muestran en la figura:



3. Al presionar OK, se realizará la conexión hacia la base de datos, y aparecerá en la lista de conexiones.



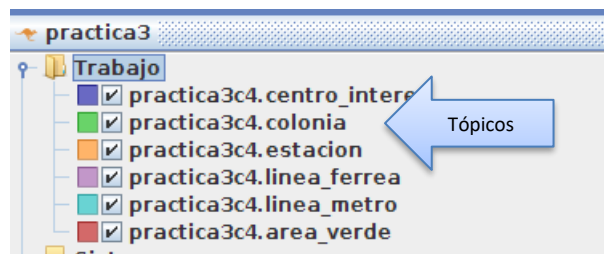
4. Ejecución de sentencias SQL.

Con OpenJump es posible ejecutar sentencias SQL y mostrar los resultados de forma gráfica. El único requisito, es que la sentencia SQL solo debe regresar una columna y debe corresponder con algún ADT. Observar adicionalmente, es necesario indicar el nombre del esquema ("practica3").

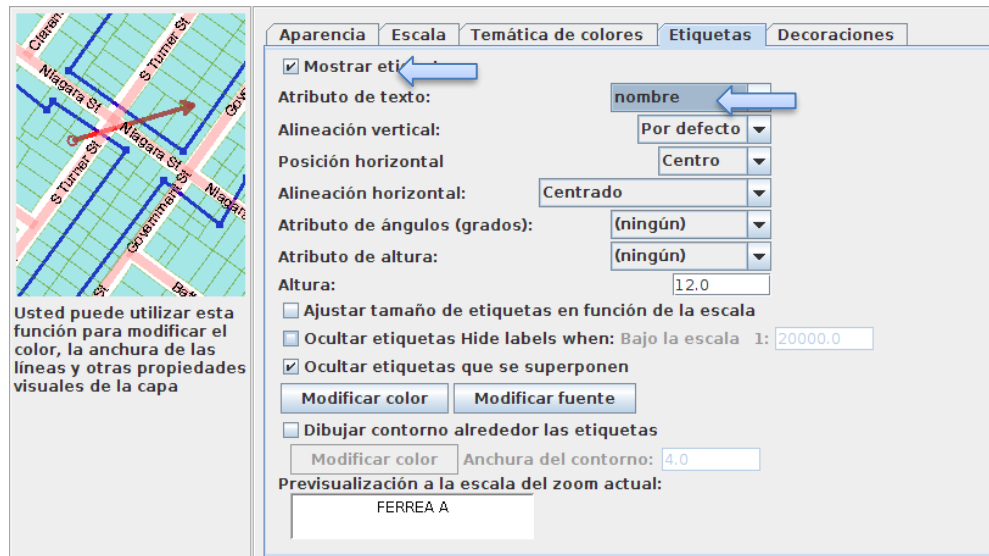
En la imagen siguiente se muestra una consulta que dibuja las geometrías de las colonias. **Ejecutar** la instrucción. **C4: incluir en el reporte la imagen producida.**

5. Salvar el proyecto. Para no perder las configuraciones realizadas en OpenJump, seleccionar Archivo -> Guardar proyecto. Observar que, a cada consulta, OpenJump la maneja como una “capa temática” a la que se le asigna un nombre.
6. Carga de capas temáticas.  
En OpenJump es posible mostrar el área geográfica de análisis de forma gráfica separada por capas. Cada capa es representada por una tabla. Para cargar las capas temáticas del área de análisis, ejecutar las siguientes instrucciones:
  - Dentro del panel izquierdo, hacer clic derecho en la carpeta llamada “Trabajo”, seleccionar la opción “agregar capa de almacén”, proporcionar los datos como se muestra a continuación, no olvidar especificar el nombre del esquema.

- Aplicar este mismo procedimiento para cada uno de los tópicos. Al final, aparecerán todos los tópicos similar a la siguiente imagen:



- Observar que del lado derecho se muestran los diagramas que pertenecen a cada capa. Verificar que estén seleccionadas todas. **C5: incluir en el reporte el diagrama generado.**
- Al diagrama se le puede anexar etiquetas que muestren algún atributo alfanumérico del tópico. Para habilitar las etiquetas, hacer clic derecho sobre el tópico (ver figura anterior), seleccionar la opción estilo->modificar estilo, pestaña Etiquetas. Habilitar las y seleccionar el nombre del atributo a mostrar:



### 1.2.6. Análisis espacial.

Para cada uno de los siguientes enunciados, generar una sentencia SQL que atienda el requerimiento solicitado. **C6: Incluir en el reporte** la sentencia SQL y la salida (pantallazo) de la consulta.

Notas:

- En ningún caso emplear ST\_INTERSECTS.
- Evitar el uso de WITH a menos que sea indicado o donde se justifique su empleo (reutilización de consultas).

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.
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

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. Mostrar las etiquetas con los nombres de los objetos.
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.
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)

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 tópicos de las líneas férreas para observar que efectivamente la línea férrea se extenderá.

Notas: no hacer uso del punto asociado a la estación FF1. Se debe determinar el punto de bifurcación como el punto final del primer segmento de recta del multilinestring que define a la línea férrea. Para crear una poli línea a partir de 2 puntos se puede emplear la función st\_makeLine.

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.
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.
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 el número asignado a la geometría. Recomendación, emplear `ST_DUMP` y el arreglo `path` para extraer el índice de la geometría.

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.

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.
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 línea. Ordenar los resultados por el nombre de la estación, no emplear `WITH` ni subqueries, emplear sintaxis estándar.

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. Ordenar por el nombre de las áreas verdes.

Ejemplo:

Nombre colonia	Nombre área verde	Área
AGUIRRE	AV2	xx
SUAREZ	AV3	xx

### 1.3. CONTENIDO DEL REPORTE

- Introducción
- Objetivo
- Desarrollo de la práctica:
  - C1: Sentencias SQL que agregan la columna `geo` con su ADT asignado.
  - C2: Inclusión de sentencias SQL con los constraints espaciales.
  - C3: 5 sentencias `INSERT` de ejemplo con su correspondiente geometría
  - C4: Mapa de OpenJump que muestre las geometrías de las colonias. Agregar etiquetas de nombre.
  - C5: Mapa de OpenJump con todos los tópicos seleccionados. Agregar etiquetas de nombres.
  - C6: Análisis espacial, sentencia SQL y resultado para cada ejercicio, diagrama en OpenJump donde se solicite.
- Conclusiones, comentarios, recomendaciones.
- Bibliografía.

### 1.4. DOCUMENTACIÓN DE POSTGIS PARA APOYO ADICIONAL:

<http://postgis.net/docs/manual-2.1/>

<http://postgis.net/docs/manual-2.1/reference.html>



## 1.5. DIAGRAMAS ZONA GEOGRÁFICA:

