



Proyecto YuppGIS

Procesador de Consultas

Página Web

<http://yuppgis.googlecode.com/>

Estudiantes

Emilia Rosa

Jorge Davison

Martin Taruselli

German Schnyder

Bajo el esquema de Paquete Premium, YuppGIS brinda flexibilidad a la hora de trabajar con conexiones a base de datos diferentes (geográfica y no geográfica).

Para permitir esa flexibilidad, se debió implementar un procesador de consultas orientadas a objetos del dominio, capaz de trabajar con dichas conexiones. La principal dificultad se encuentra en el procesador a la hora de realizar *joins* entre tablas diferentes, en bases de datos diferentes, con conexiones diferentes.

Para superar esta dificultad, se diseña un algoritmo de resolución de consultas que depende de la consulta en sí misma. La lógica de este algoritmo toma en cuenta los atributos que se quiera seleccionar, atributos por los que se aplican las condiciones y la cantidad de conexiones disponibles.

Si las conexiones a los motores de base de datos (con extensión geográfica o sin) son hacia el mismo motor-base de datos, las consultas se interpretan y se delegan los *joins* necesarios al motor de base de datos.

En caso que las conexiones a motores de base de datos (con extensión geográfica o sin), son hacia distintos motores y/o base de datos, las consultas se interpretan y se realizan los *joins* necesarios en memoria.

La interpretación de consultas es la parte más importante del procesador de consultas, ya que su objetivo es la creación de nuevas consultas SQL a partir de una consulta orientada a objetos del dominio.

Si se cuenta con una sola conexión a base de datos, entonces el procesador desglosa dicha consulta y genera una nueva consulta SQL, ejecutándose sobre el motor de base de datos.

Si se cuenta con conexiones distintas a bases de datos, se analizan los atributos a seleccionar geográficos y no geográficos, creando una nueva consulta SQL con todos aquellos atributos y condiciones no geográficas. En dichas consultas se agregan, además de los atributos a seleccionar, el identificador de la geometría asociada a dichos elementos, debido a que dicha geometría se encuentra en una tabla distinta. Luego, para cada selección de atributos geográficos sobre elementos del dominio distintos, se genera una nueva consulta SQL, en la cual se aplica como condición extra, además de las condiciones geográficas especificadas en la consulta original, la condición de que la geometría tenga como identificador alguno de los identificadores que se obtuvieron como resultado en la primera consulta SQL. Para finalizar, se realiza el *join* de los resultados obtenidos en memoria, generando el resultado final.

A continuación se presenta un ejemplo, desglosando en pasos el procedimiento del procesador de consultas.

Consulta original:

```
$q = new GISQuery();
$q->addProjection('p', 'ubicacion', 'ubicacion_de_p');
$q->addProjection('p', 'nombre', 'nombre_de_p');
$q->setCondition(Condition::_AND()
    ->add(Condition::EQ('p', 'nombre_de_p', 'Juan'))
    ->add(GISCondition::EQGEO('p', 'ubicacion_de_p', new Point(-56.181948, -34.884621)))
);
$q->addFrom(Paciente::getClassName(), 'p');
```

Procedimiento que ejecuta el procesador de consultas:

1. Interpretación de consultas orientadas a objetos, descomponiendo y creando consultas SQL.

Nombre	SelectItem('nombre_de_p') Reference: nombre
ubicacion_id	SelectItem('ubicación_de_p') Reference: ubicacion_id

2. Ejecución de consultas SQL.

- a. Ejecución de consultas geográficas basadas en los criterios de filtrado, obteniendo un 'resultado_condicion'

```
SELECT u.id FROM tabla_paciente_ubicacion_geo u WHERE ST_Equals(u.the_geom ,
AsText('Point(-56.181948 -34.884621)'));
```

- b. Ejecución de consultas no geográficas basadas en la obtención de datos del objeto, obteniendo un 'resultado'

```
SELECT p.nombre, p.ubicacion_id FROM tabla_paciente p WHERE p.nombre = 'Juan' AND
p.ubicacion_id IN ({resultado_condicion});
```

3. Ejecución de consultas SQL para obtención de datos geográficos, 'resultado_gis'.

```
SELECT u.id, u.the_geom, u.uiproperty FROM tabla_paciente_ubicacion_geo u WHERE u.id =
{resultados}[0] OR u.id = {resultados}[1] OR ... OR u.id = {resultados}[n];
```

Donde **n** es la cantidad de elementos en "resultado". En dicha consulta se genera los objetos geográficos, por lo que el resultado de esta consulta serán objetos Point con sus propiedades en vez de atributos por separados.

4. Generación de resultado final en base al siguiente algoritmo.
 - a. Por cada elemento en 'resultado'
 - i. Por cada SelectListItem o GISFunction de la consulta original
 1. Se unen los resultados de 'resultado' y 'resultado_gis', correspondientes al elemento SelectListItem o GISFunction de la consulta original.

ubicacion_de_p	nombre_de_p
Point(-56.181948, -34.884621)	Juan