



Geoinquietos Valencia

Marzo 2012

Novedades

PostGIS 2 - PostgreSQL 9.1

José Carlos Martínez



PostGIS 2 y PostgreSQL 9.1

- Novedades PostGIS 2
- Instalación (PG 2 y Postgres 9.1)
- **Gestión de tablas espaciales**
- Vistas de metadatos (PG 2*)
- Dependencia funcional (Postgres 9.1)
- **Raster** (PG 2*)
- **Topología persistente** (PG 2*)
- ¿Qué hay que cambiar en nuevos drivers?
 - *Comentar trabajo en versiones previas

• • • • • • • • • •



Novedades PostGIS 2

- Todas las funciones utilizan el prefijo 'ST_' (excepciones `geometry_type`, `addgeometrycolumn`, etc.)
- Funcionalidad vectorial 3D ampliada (no deja aún de ser un SIG de 2.5 dimensiones)
- Indexación espacial 3D y KNN (KNN aún por mejorar)
- Vistas de metadatos en lugar de tablas (*geography_columns* en 1.5)
- Posibilidad de *Typemod* en lugar de restricciones *Check*
- Raster
- Topología persistente (importante especialmente para modelos cartográficos)
- Algunos cambios: SRID = 0 (defecto), funcionamiento algunas funciones (`ST_StartPoint..`), WKB, WKT





Instalación (PG 2 y Postgres 9.1)

- Antes->

- `psql -U postgres -f /ruta/a/ficheros/postgis.sql`
- `psql -U postgres -f /ruta/a/ficheros/rtpostgis.sql`
- `psql -U postgres -f /ruta/a/ficheros/postgis_topology.sql`

- Ahora:

- --Vectorial y raster

Create extension postgis;

- --Topología persistente

Create extension postgis_topology;

- Directorio *extensions* dentro del directorio de instalación de PostgreSQL

- Siempre se pueden ejecutar los ficheros *postgis.sql*, *rtpostgis.sql* y *postgis_topology.sql* de forma manual si no se tiene PG 9.1



Gestión de tablas espaciales

- Sigue permitiendo crear tablas donde las restricciones de las geometrías son controladas por cláusulas CHECK de tabla.
- El comportamiento buscado es sin embargo prescindir de las restricciones CHECK utilizando la parametrización de las geometrías con *typemod* (comportamiento por defecto)
- 1.- *Create table ttmm (gid serial, geom geometry (polygon, 25830));*
- 2.- Esto es igual que realizar:

Create table ttmm (gid serial);

Select addgeometrycolumn ('ttmm','geom',25830,'POLYGON', 2, true);

Ultimo argumento es true por defecto que significa:

Utiliza *typemod* en lugar de *CHECKs*

1) es mucho más cómodo pero quizás: es menos compatible OGC al dejar de utilizar *addgeometrycolumn*? → hay que decir sin embargo que *addgeometrycolumns*, etc eran métodos aconsejados y no obligatorios.



Vistas de metadatos

- Geometry_columns es una vista a partir del catálogo de PostgreSQL donde averigua que restricciones CHECK o los parámetros typemod de la columna de geometría.
- Muchas ventajas:
 - No hay que ocuparse de sincronización
 - Funciones tipo probe_geometry_columns, populate_geometrycolumns incluso addgeometrycolumn, dropgeometrycolumn, dropgeometrytable no se necesitan.
- Para un programas externo (SIG escritorio) no hay problema, incluso si se trata de modificar la vista se han definido unas reglas (create rule) sobre la vista para que una orden Insert, Delete, etc. NO haga nada.
- Y las VISTAS espaciales (con CAST)→ create view vistaespacial (gid, geom) as select gid, st_transform(geom, 25830)::geometry(polygon, 25830) from tablaoriginal;



Cambio comportamiento no lógico?

- ST_StartPoint (multi) → NULL (antes ok)
- ST_EndPoint (multi) → NULL (antes ok)
- ST_NumGeometries (single) → 1 (antes null)
- Comportamiento no OGC/SQL MM





Dependencia funcional PG 9.0

```
s1#= insert into erase1 (tema, grupo, geom)
select tema, grupo, geom from
( select max(tema) as tema, max(grupo) as grupo,
  stx_extract(st_difference (
    (st_Accum(s.geom))[1],
    st_union(n.geom)),2 ) as geom
from suelos s, nucleos n
where s.geom && n.geom and
  st_relate (s.geom, n.geom, 'T*****') group by s.gid
) as tabla where geom is not null;
```




Indexación 3D

```
create index p3d_gist3d on p3d  
  using gist (geom gist_geometry_ops_nd);
```

```
select count(*) from p3d  
  where geom &&& st_geomfromewkt ('LINESTRING  
(500500 4500200 50, 500800 4500700 51)');
```

Se puede mejorar un orden de magnitud el rendimiento de la indexación 2D fácilmente.

- Sobre todo se nota la ventana en búsquedas con cajas de
- amplia extensión planimétrica y con un rango de Zs pequeño)



Indexación KNN 1/2

KNN (*K Nearest Neighbors*)

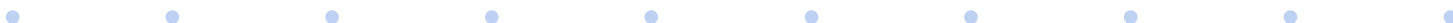
PostGIS 2.0 implementa dos nuevos operadores:

`'geom1 <#> geom2'` (distancia entre cajas)

`'geom1 <-> geom2'` (distancia entre centroides)

Para que se utilice los índices espaciales hay que cumplir:

Una de las dos geometrías utilizadas con el operador (*geom1* o *geom2*) debe ser constante → Limitación





Indexación KNN 2/2

```
s1=# select nombre
      from meteoche m
      order by m.geom <#> ST_GeomFromText('POINT(530660
4744550)')
      Limit 5;
```

```
s1=# select nombre from meteoche m
      where st_dwithin (m.geom,
          ST_GeomFromText('POINT(530660 4744550)', 23030), 10000)
      order by st_distance (m.geom,
          ST_GeomFromText('POINT(530660 4744550)', 23030))
      limit 5;
```



Comandos análisis 3D

- Se han renombrado al prefijo ST_3D
Ej: ST_Length3D → ST_3DLength
- Muy pocas operaciones que soportan 3D de verdad, las que hay están implementadas por PostGIS ya que ni GEOS ni JTS soportan nada de análisis espacial 3D.
- Algunos ejemplos de análisis 3D:
 - ST_3DIntersection (solo con puntos y/o líneas)
 - ST_3DClosestPoint
 - ST_3DDWithin (calcula plano medio en entidades superficiales)
- ¿Cuándo GEOS/JTS implementarán análisis espacial vectorial 3D?

• • • • • • • • • •



Otras nuevas características

- Funciones extra: ST_MakeValid, ST_Snap, ST_RemoveRepeatedPoints, ST_ConcaveHull, ST_Split, etc. (solo algunas más si no contamos raster y topología persistente).
- Algunos nuevos métodos que funcionan con coordenadas geográficas (utilizando cálculos sobre el elipsoide) → ST_Area, ST_DWithin
- TIGER Geocoder (no incluido dentro del core → EEUU) aunque mejorado comparado con versiones anteriores.





Geometrías curvas

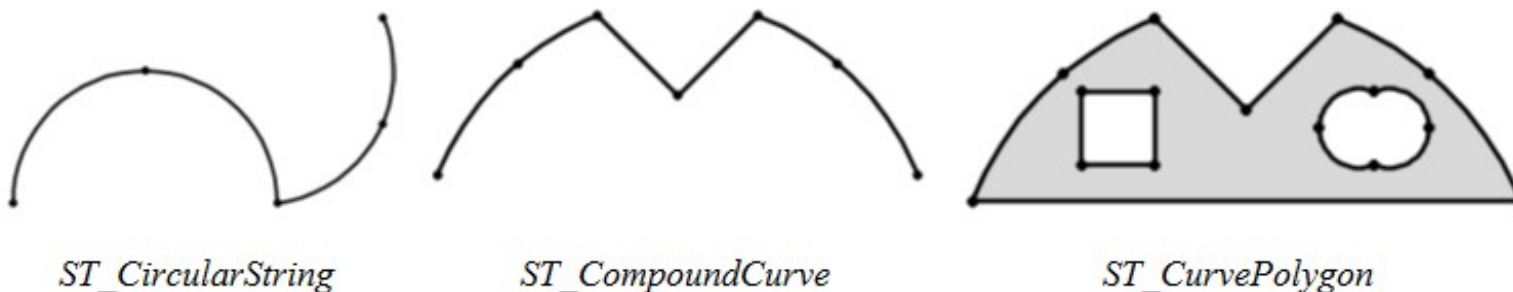


Figura 65 Tipos de geometrías curvas

- La mayoría del análisis espacial no es soportado por las geometrías curvas:
- Es necesario utilizar antes del análisis: `ST_CurveToLine` (poco sentido)
- Poco a poco irán introduciendo más funcionalidad. GEOS/JTS no soportan dichas geometrías.
- No soportados por ningún SIG de escritorio actualmente



PostGIS Raster 1/3

- Con algunas decenas de funciones aporta unas posibilidades enormes al análisis espacial raster y al mixto vectorial-raster
- Cargador raster2pgsql, ya no es python se han eliminado las dependencias.
- Muchas de las funcionalidades las realiza GDAL, biblioteca en la cuál está basado.
- GvSIG → jsanz.github.com/slides/foss4g-2011/gvsig_pgraster.html
- http://trac.osgeo.org/postgis/attachment/wiki/WKTRaster/PierreRacine_FOSS4G-2011.pdf
- <http://trac.osgeo.org/postgis/wiki/WKTRaster>
- GvSIG 2 (Nacho Brodin) → ¿Driver quizás ya en el trunk?



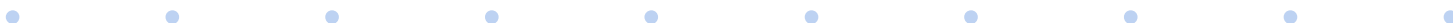
PostGIS Raster 2/3 - Ventajas

- Increíble versatilidad para la realización de análisis espaciales raster, raster-vectoriales.
- *Vectorización*
- Rasterización
- Funciones de vecindad (incluso con funciones personalizadas)
- Algebra de mapas
- Recorte, unión, intersección, raster-raster, raster-vectorial.
- Reclasificación
- Funciones de ayuda (Slope, Aspect, Hillshade, etc.)
- GvSIG, QGIS ya soportan la visualización al menos
- GDAL
- Todas las que da un SGBD (SQL, multiusuario, transacciones, etc.)



PostGIS Raster 3/3 — Desventajas actuales ?

- Los métodos que no funcionan a nivel de *tiles* sino a nivel pixel tienen problemas de rendimiento.
- Ciertas operaciones de análisis, por ejemplo de vecindad (slope, aspect, etc) por ahora necesitan agrupar todos los tiles en un único raster para obtener resultados correctos (la franja de los bordes de los tiles no presentan valores correctos)
- Si únicamente vamos a visualizar los rasters (finalidad única servir WMS) quizás no vale la pena meterlos en PostGIS ?
-





PostGIS Raster — Ejemplo suavizado integral de una capa y nuevo teselado

```
create table suavi (rid serial, rast raster);

with tabla as (
  select st_mapalgebrafctngb
    (rast, 1, '32BF', 4, 4, 'st_mean4ma(float[][],text,text[])'::regprocedure, 'ignore', NULL) as rast
  from (select st_union(rast2) as rast from dem1) as tabla1
)

insert into suavi (rast) select st_clip (tabla.rast, st_envelope (dem1.rast2), 0, true)
  from dem1, tabla where st_intersects (dem1.rast2, tabla.rast);
```



Topología persistente, por fin!

- Sigue modelo SQL/MM o ORACLE Spatial
- Es bastante más que modelo arco-nodo tipo ArcInfo Workstation o Grass
- Ideal y quizás única forma rigurosa para representar modelos cartográficos tipo BTA (Base topográfica armonizada).
- Pon un ejemplo gráfico en PIZARRA!!
- Quizás en la versión 2.0 aún no demasiado estable
- Futuro-> Análisis espacial utilizando la topología persistente
- Editores de topología en los SIG de escritorio.
- Curiosidad: <http://www.es.pledgebank.com/postgistopology>





Tipos TIN y PolyhedralSurface

- Se trata de geometrías que no se pueden estudiar por separado sino que tienen una clara dependencia con las geometrías que les rodean
- Hasta ahora muy muy pocas funcionalidades, de análisis espacial → GEOS/JTS (solo JTS tiene TIN).
- Futuro->Quizás la implementación mediante topología persistente en lugar de en un sistema de simple features es más lógica?

-
-
-





¿Qué hay que cambiar en nuevos drivers?

- Poner funciones con prefijo ST_
- Tener en cuenta que ST_AsBinary y ST_AsText devuelven la Z (antes solo XY).
- No es necesario agregar ni borrar registros de geometry_columns, si se hace no pasa nada porque hay rules definidas en la vista.
- Comportamiento de ciertas geometrías: ST_StartPoint, ST_NumGeometries, etc.
- A ser posible no utilizar el jar de JDBC de PostGIS, generalmente tanto gvSIG/kosmo etc. solo lo utilizan por el tipo BBOX->Se puede eliminar su dependencia utilizando funciones ST_Envelope, ST_X, ST_Y
- SRID = 0 (defecto) en lugar de -1



Conclusiones

- Lo mejor: typemod, vistas de metadatos, raster, topología persistente
- Las funciones 3D, geodésicas, curvas, geometrías superficiales, tolerancias, nuevas versiones de SFA -> NO son soportadas por JTS aún → Esto es una gran limitación no solo para PostGIS sino para todo el software open source Geoespacial.
- JTS/GEOS están quedándose anticuadas → se dedica mucho esfuerzo a otros proyectos GeoTools, PostGIS, gvSIG, etc. la mayoría de los cuáles se apoya en JTS para el análisis espacial.
- Con el Raster y la topología persistente PostGIS está mucho más cerca de Oracle Spatial



Geoinquietos Valencia Marzo 2012

Gracias por la atención

José Carlos Martínez