

Kolme pientä PostGIS-esimerkkiä

Jonne Savolainen Janne Upla

19. maaliskuuta 2015

- PostGIS on PostgreSQL:n laajennosrajapintaa käyttävä lisäosa, joka toteuttaa Simple Features Specification -määrittelyn
(<http://www.opengeospatial.org/standards/>)
- Projekti perustettu vuonna 2001
- Tukee tasogeometrioita (GEOMETRY), sferoidigeometrioita (GEOGRAPHY) ja rasteritiiliä (RASTER)
- Kevyt arkkitehtuuri, satoja funktioita
- Perusteellinen esittely:
<http://workshops.boundlessgeo.com/postgis-intro/>

Esimerkki 1: Postinumeroaineiston yleistäminen

Yleistysprosessi (Janne Upla)

- Perustetaan PostGIS-kanta `postgis_topology`-laajennoksella
- Ladataan taulu Tilastokeskuksen WFS-palvelusta tietokantaan `ogr2ogr`-työkalulla (<http://www.gdal.org/>)
- Muodostetaan topologinen malli
- Yleistetään rajaviivageometrioita topologisen mallin avulla
- Vertailun vuoksi yleistetään polygonit ilman topologista mallia

Esimerkki 1: PostGIS-kannan perustaminen

Luodaan ensin tyhjä Postgres-kanta (versio ≥ 9.1), omistaja fipug.

```
$ createdb -U postgres -O fipug fipug
```

```
$ psql -U postgres -d fipug
```

```
CREATE EXTENSION postgis;
```

```
CREATE EXTENSION postgis_topology;
```

```
GRANT ALL ON SCHEMA topology TO fipug;
```

```
GRANT ALL ON ALL TABLES IN SCHEMA topology TO fipug;
```

```
GRANT ALL ON ALL SEQUENCES IN SCHEMA topology TO fipug;
```

```
CREATE SCHEMA statfi;
```

```
ALTER DATABASE fipug
```

```
SET search_path = statfi, public, topology, pno_meri_topo;
```

Esimerkki 1: Ladataan aineisto tietokantaan

```
$ ogr2ogr -overwrite -f PostgreSQL\  
  PG:"user=fipug host=localhost port=5433 dbname=fipug"\  
  WFS:http://geo.stat.fi/geoserver/postialue/postialue:pno_meri/wfs\  
  -nln pno_meri postialue:pno_meri  
  
-- Tarkistetaan, että tulos on järkevä.  
SELECT sum(ST_Area(wkb_geometry)) / 1000000.0 AS km2  
FROM statfi.pno_meri;  
      km2  
-----  
      390815.261762989  
(1 row)
```

Esimerkki 1: Topologisen mallin luominen

```
SELECT CreateTopology('pno_meri_topo', 3067);
      createtopology
```

```
-----
              1
(1 row)
```

```
SELECT AddTopoGeometryColumn('pno_meri_topo',
      'statfi', 'pno_meri', 'topogeom', 'MULTIPOLYGON');
      addtopogeometrycolumn
```

```
-----
              1
(1 row)
```

```
SET client_min_messages TO WARNING;
```

```
UPDATE statfi.pno_meri
```

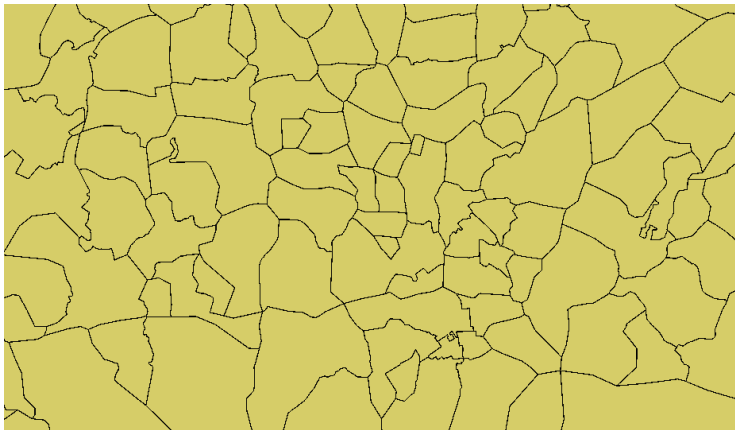
```
SET topogeom = toTopoGeom(wkb_geometry, 'pno_meri_topo', 1);
```

Esimerkki 1: Yleistäminen

```
-- Ei-topologinen yleistys
ALTER TABLE statfi.pno_meri
  ADD COLUMN geom_bork geometry(Multipolygon,3067);
UPDATE statfi.pno_meri
  SET geom_bork = ST_Multi(
    ST_SimplifyPreserveTopology(wkb_geometry, 500)
  );

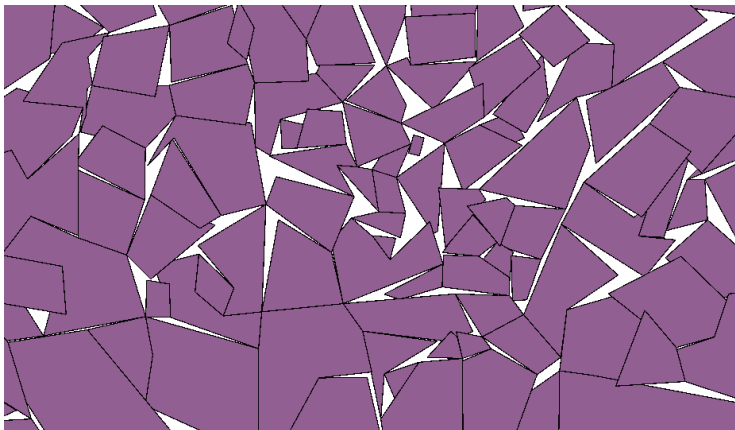
-- Topologinen yleistys
ALTER TABLE statfi.pno_meri
  ADD COLUMN geom_simp geometry(Multipolygon,3067);
UPDATE statfi.pno_meri
  SET geom_simp = ST_Simplify(topogeom, 500);
```

Esimerkki 1: Alkuperäiset alueet (wkb_geometry)



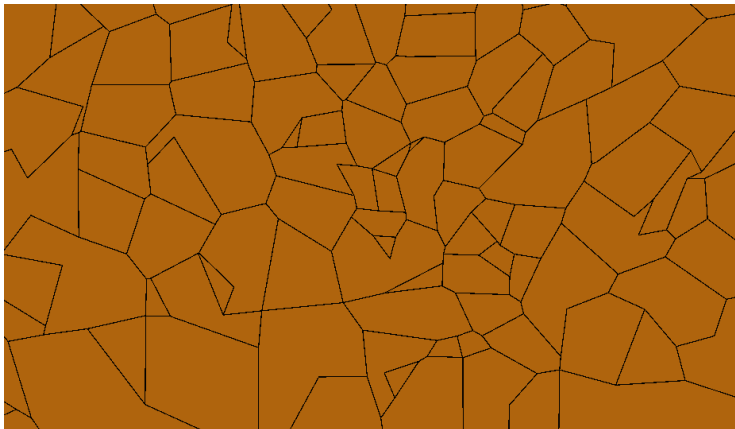
Kuva: Alkuperäiset postinumeroalueet ilman yleistystä.

Esimerkki 1: Ei-topologinen yleistys (geom_bork)



Kuva: Alueet yleistettynä yksittäin ilman naapurialueiden huomioimista.

Esimerkki 1: Topologinen yleistys (geom_simp)



Kuva: Alueet yleistettynä yhteisten rajaviivojen avulla (postgis_topology)

Esimerkki 2: Rekursio postinumeroalueilla

```
-- 00100 Postinumeroalueen naapureita
WITH RECURSIVE t(ogc_fid, the_geom, n, visited, cycle) AS (
    SELECT pno_meri.ogc_fid, pno_meri.wkb_geometry,
           1, ARRAY[]::int[], false
    FROM statfi.pno_meri
    WHERE pno_meri.posti_alue = '00100'
    UNION ALL
    SELECT pno_meri.ogc_fid, pno_meri.wkb_geometry,
           t.n + 1, t.visited || t.ogc_fid, t.ogc_fid = ANY(t.visited)
    FROM statfi.pno_meri, t
    WHERE t.the_geom && pno_meri.wkb_geometry
           AND ST_Overlaps(ST_Boundary(t.the_geom),
                           ST_Boundary(pno_meri.wkb_geometry))
           AND t.ogc_fid != pno_meri.ogc_fid
           AND NOT ARRAY[pno_meri.ogc_fid] <@ t.visited
           AND t.n < 6
           AND NOT cycle
)
SELECT DISTINCT n, ST_Transform(the_geom, 900913) FROM t;
```

Esimerkki 3: KNNGIST-haku OSM-aineistolla

- OSM-ainesto haettu maittain: <http://download.geofabrik.de/>
- Tietokantaan vienti: osm2pgsql
- Partitioitu maittain perintää käyttäen
- Luotu GIST-indeksi geometriakentälle "way" ja GIN-indeksi hstore-kentälle "tags"
- Aineiston koko:

```
SELECT pg_size_pretty(  
    sum(pg_total_relation_size(schemaname || '.' || tablename)))  
FROM pg_tables WHERE tablename LIKE 'osm_%_point';  
pg_size_pretty  
-----  
2601 MB  
(1 row)
```

Esimerkki 3: KNNGIST-haku OSM-aineistolla

```
--  
-- 100 lähintä pubia kursorin paikasta  
--  
SELECT way, ST_Distance(ST_SetSRID(::cp)::geometry, 900913), way)  
FROM osm.osm_point  
WHERE amenity = 'pub'  
ORDER BY osm_point.way <-> ST_SetSRID(::cp)::geometry, 900913)  
LIMIT 100;  
  
--  
-- Karaokella (?-operaattori hyödyntää tags-kentän GIN-indeksiä)  
--  
SELECT way, tags -> 'karaoke' AS karaoke,  
       ST_Distance(ST_SetSRID(::cp)::geometry, 900913), way)  
FROM osm.osm_point  
WHERE amenity = 'pub'  
      AND tags ? 'karaoke'  
ORDER BY osm_point.way <-> ST_SetSRID(::cp)::geometry, 900913);
```

Mandelbrot-fraktaali

```
--
-- Mandelbrot SQL:llä
-- (vaatii Common Table Expression -tuen).
--
-- http://wiki.postgresql.org/wiki/Mandelbrot\_set
--
WITH RECURSIVE x(i) AS (
    VALUES(0)
UNION ALL
    SELECT i + 1 FROM x WHERE i < 101
),
Z(Ix, Iy, Cx, Cy, X, Y, I) AS (
    SELECT Ix, Iy, X::float, Y::float, X::float, Y::float, 0
    FROM
        (SELECT -2.2 + 0.031 * i, i FROM x) AS xgen(x,ix)
    CROSS JOIN
        (SELECT -1.5 + 0.031 * i, i FROM x) AS ygen(y,iy)
    UNION ALL
    SELECT Ix, Iy, Cx, Cy, X * X - Y * Y + Cx AS X, Y * X * 2 + Cy, I + 1
    FROM Z
    WHERE X * X + Y * Y < 16.0
    AND I < 27
), Zt (Ix, Iy, I) AS (
    SELECT Ix, Iy, MAX(I) AS I
    FROM Z
    GROUP BY Iy, Ix
    ORDER BY Iy, Ix
)
SELECT ST_Translate(ST_Buffer(
    ST_MakePoint(3000*Ix, 3000*Iy),
    20 + I*25),
    -150000, 7100000, 0)
FROM Zt;
```

Mandelbrot-fraktaali

