PostGIS: Up and Running

Regina Obe and Leo Hsu

Buy our books! at http://www.postgis.us

PGOpen promo codes for 44% off PostGIS In Action 2nd Edition
postgrescft

Tutorial material at http://www.postgis.us/pgopen2014

Agenda

Preliminaries

Geometry

Spatial Reference Systems

Geography

Geocoding

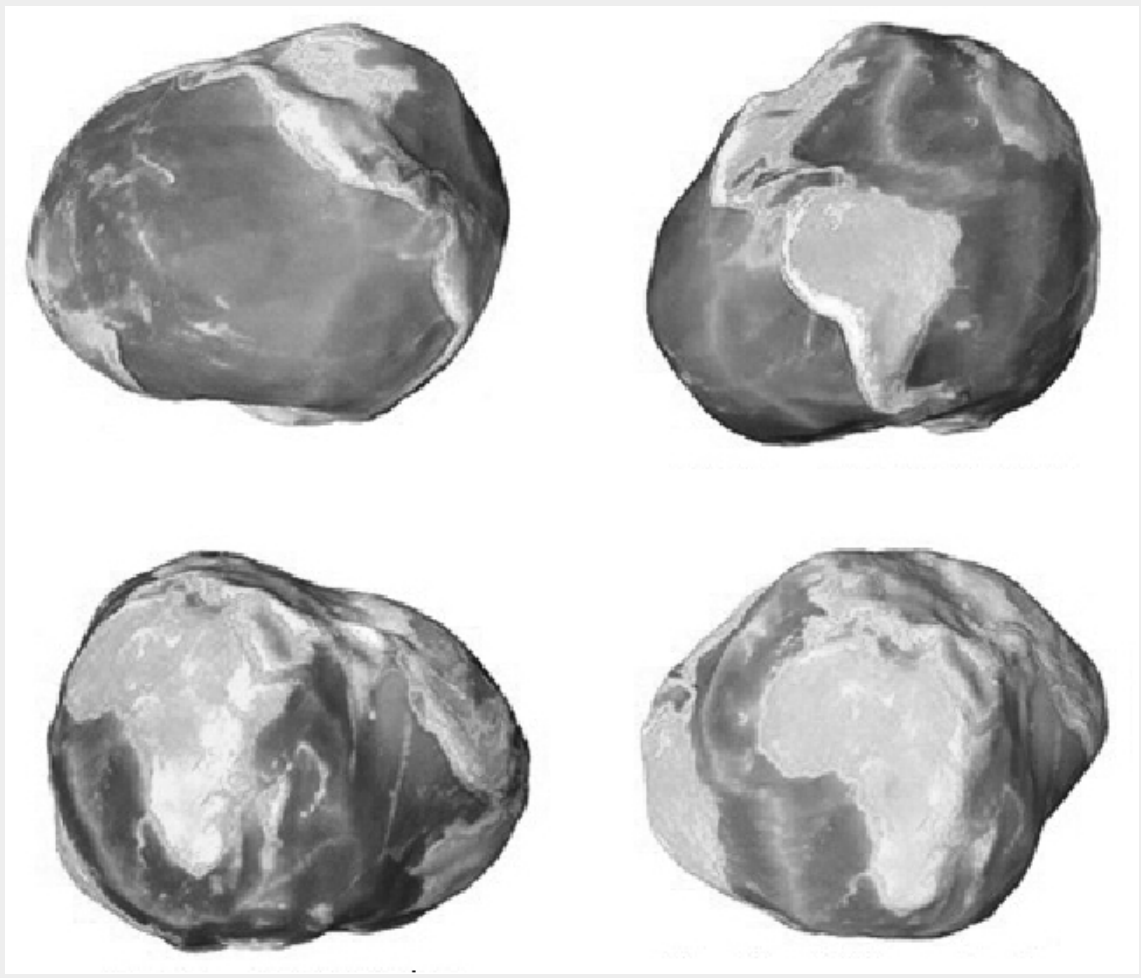
3D Geometries

Tantalizing SQL Constructs

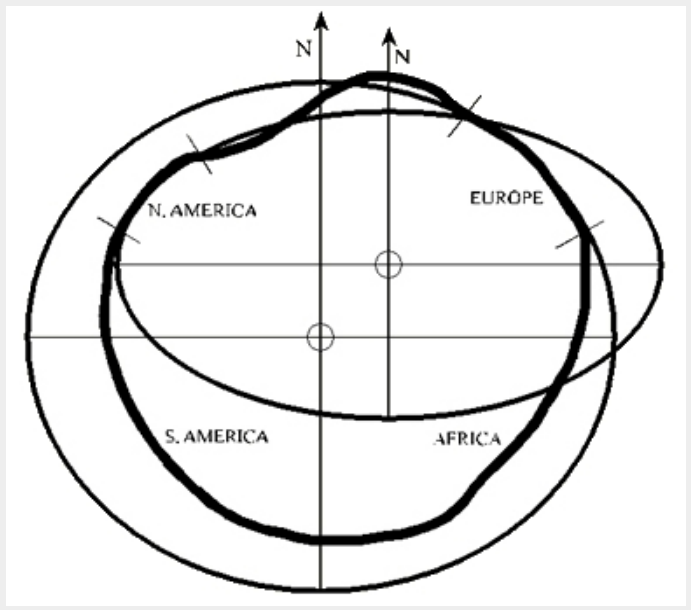
Importing Data

Tools of the Trade

Geoid



Ellipsoid



OS2PGSQL and other OpenStreetMap loaders

- Osm2pgsql http://wiki.openstreetmap.org/wiki/Osm2pgsql is one of several tools not packaged with PostGIS but useful for loading OSM data into PostGIS format. Supported on all platforms and was one of the first
- Imposm http://imposm.org - another popular for Linux/Mac OSX - no binaries for windows. Many say it is faster than osm2pgsql.
- ogr2ogr Part of GDAL toolkit. Binaries available for many platforms and often already installed if you installed PostGIS on Linux.

3D Surfaces

We are using PostGIS `ST_AsX3D` function combined with http://www.x3dom.org to render some of these in HTML

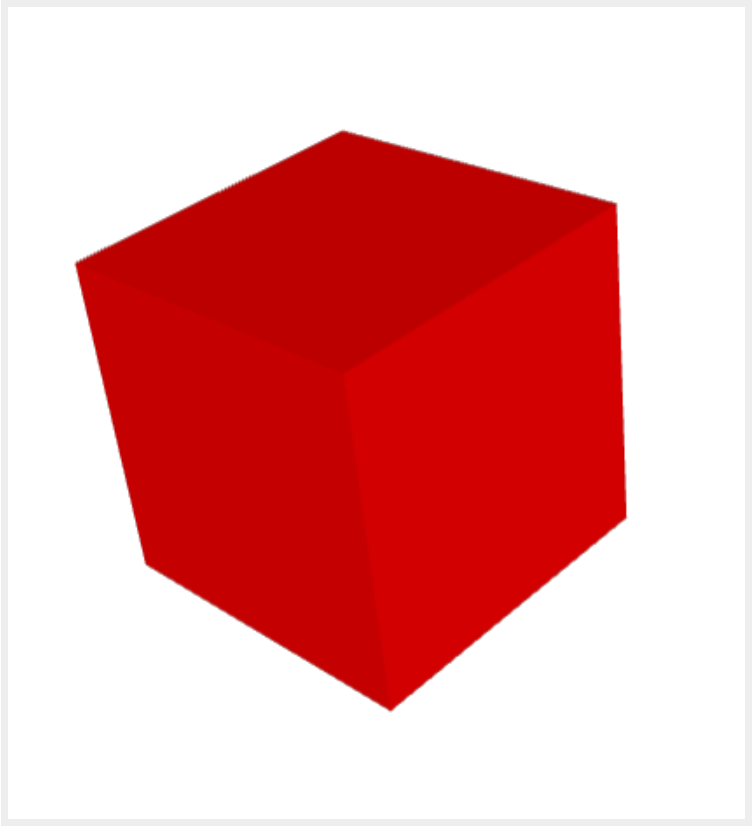
Warning 3D stuff is still not very tested yet.

Closed Polyhedral Surface

```
SELECT ST_GeomFromText('POLYHEDRALSURFACE Z(
   ((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0))
 , ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0))
 , ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0))
 , ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0))
 , ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0))
 , ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)) )');
```

Closed Polyhedral Surface In X3D

```
SELECT ST_AsX3D('POLYHEDRALSURFACE Z(
   ((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0))
 , ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0))
 , ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0))
 , ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0))
 , ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0))
 , ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)) )'::geometry);
```



Output Closed PolyhedralSurface with X3D

```
SELECT '<shape>
 <appearance><material ambientintensity="0.200" containerfield="material" shininess="0.200" diffusecolo
ST_AsX3D('POLYHEDRALSURFACE Z(
   ((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0))
 , ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0))
 , ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0))
 , ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0))
 , ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0))
 , ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)) )'::geometry) || '</shape>';
```

Closed TIN

```
SELECT 'TIN(
            ((0 0 0, 1 0 0, 0 1 0, 0 0 0)),
            ((0 0 0, 1 0 0, 0 0 1, 0 0 0)),
            ((0 0 0, 0 0 1, 0 1 0, 0 0 0)),
            ((0 0 1, 1 0 0, 0 1 0, 0 0 1))
            )'::geometry;
```

Closed TIN with X3D

```
SELECT '<shape><appearance>
  <material ambientintensity="0.200" containerfield="material" shininess="0.200" diffusecolor="0.8 0 0.
    || replace(ST_AsX3D('TIN(
    ((0 0 0, 1 0 0, 0 1 0, 0 0 0)),
    ((0 0 0, 1 0 0, 0 0 1, 0 0 0)),
    ((0 0 0, 0 0 1, 0 1 0, 0 0 0)),
    ((0 0 1, 1 0 0, 0 1 0, 0 0 1))
            )'::geometry),
   '<IndexedTriangleSet', '<IndexedTriangleSet solid="false" ') || '</shape>';
```

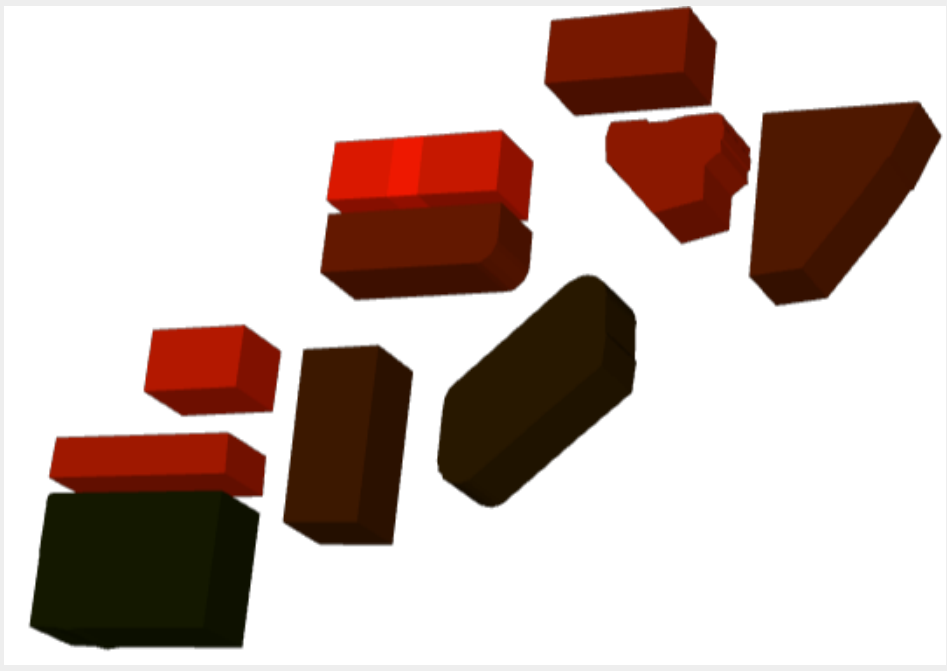
Road network in X3D

```
SELECT string_agg('<shape>
 <appearance><material ambientintensity="0.200" containerfield="material" shininess="0.25" emissivecolo
</appearance>' ||
ST_AsX3D(ST_Force_3D(geom)) || '</shape>', '') As x3d
FROM (SELECT row_number() over(ORDER BY ST_Length(way) DESC) As rn,
 name, count(*) OVER() As cnt, way As geom
 FROM po.planet_osm_line
   WHERE ST_Intersects(ST_Transform(
    ST_Buffer(
       ST_Point(-87.627,41.8819)::geography,
       100)::geometry,900913
        ) , way)  ) As f;
```

SFCGAL ST_Extrude convert 2D Buildings to 3D

```
SELECT string_agg('<shape>
 <appearance><material ambientintensity="0.200" containerfield="material" shinine
</appearance>' ||
ST_AsX3D(ST_Extrude(geom,0,0,50)) || '</shape>', '')  As x3d
FROM (SELECT row_number() over(ORDER BY ST_Area(p.way) DESC) As rn,
 name, count(*) OVER() As cnt, p.way As geom
 FROM po.planet_osm_polygon AS p
    INNER JOIN (SELECT way FROM po.planet_osm_polygon
 WHERE name ILIKE 'Trump%') As ref ON ST_DWithin(ref.way, p.way,150)
 WHERE p.building > ''
  ) As f;
```

SFCGAL ST_Extrude Output



Tiger Geocoder: Loading data

Create copy of a template record

Here I use windows. If on Linux or mac use `sh`

```
INSERT INTO tiger.loader_platform(os, declare_sect, pgbin,
        wget, unzip_command, psql, path_sep,
           loader, environ_set_command, county_process_command)
SELECT 'mydb', declare_sect, pgbin, wget, unzip_command, psql, path_sep,
           loader, environ_set_command, county_process_command
  FROM tiger.loader_platform
  WHERE os = 'windows';
```

Use your favorite editor to edit the `declare_sect`.

Generate Nation files load script

```
SELECT Loader_Generate_Nation_Script('mydb');
```

Then run the generated script from command line.

Tip 1: to reduce load skip blockgroups

If you don't need tabblocks, block groups, and tracts for statistics, disable load. These aren't used by core geocoder functionality. Disable the load.

```
UPDATE tiger.loader_lookuptables
        SET load = false WHERE table_name IN('bg', 'tabblock', 'tract');
```

Generate State load script

```
SELECT Loader_Generate_Script('{IL}'::text[], 'mydb');
```

Tip 2: Don't need whole state edit down to county

If you don't need a whole state of data, edit the generated script.

For example faces, edges, featnames -- change 17 to 17031 (if just want Cook county)

Then run the generated script from command line.

Install missing indexes and analyze

```
SELECT Install_Missing_Indexes();
```

don't forget to update stats

```
analyze verbose;
```

Geocoding

```
SELECT pprint_addy(addy),
 ST_AsText(ST_SnapToGrid(geomout,0.0001)), rating
FROM geocode('333 North Dearborn Street, Chicago, IL 60654',1);
```

```
            pprint_addy              |        st_astext        | rating
-------------------------------------+-------------------------+--------
 333 N Dearborn St, Chicago, IL 60654 | POINT(-87.6294 41.8885) |      0
```

Geocoding Intersections

Intersection Name on Lake St

```
SELECT pprint_addy(addy),
 ST_AsText(ST_SnapToGrid(geomout,0.0001)) As loc, rating
FROM geocode_intersection('Lake', 'Clark', 'IL', 'Chicago');
```

```
          pprint_addy            |          loc           | rating
---------------------------------+------------------------+--------
 101 W Lake St, Chicago, IL 60601 | POINT(-87.631 41.8857) |      5
 100 W Lake St, Chicago, IL 60661 | POINT(-87.631 41.8857) |      5
 W Lake St, Chicago, IL           | POINT(-87.631 41.8857) |      5
```

Geocoding Intersections

Intersection Name on Clark St

```
SELECT pprint_addy(addy),
 ST_AsText(ST_SnapToGrid(geomout,0.0001)) As loc, rating
FROM geocode_intersection('Clark', 'Lake', 'IL', 'Chicago');
```

```
           pprint_addy            |         loc           | rating
----------------------------------+-----------------------+--------
 200 N Clark St, Chicago, IL 60601 | POINT(-87.631 41.8857) |      5
 201 N Clark St, Chicago, IL 60601 | POINT(-87.631 41.8857) |      5
 198 N Clark St, Chicago, IL 60601 | POINT(-87.631 41.8857) |      5
 199 N Clark St, Chicago, IL 60601 | POINT(-87.631 41.8857) |      5
```

Geocoding Intersections, include zip and num matches

You get better performance if you provide Zip and max returns

```
SELECT pprint_addy(addy),
 ST_AsText(ST_SnapToGrid(geomout,0.00001)) As loc, rating
FROM geocode_intersection('Clark', 'Lake', 'IL', 'Chicago', '60601', 1);

           pprint_addy            |           loc            | rating
----------------------------------+--------------------------+--------
 200 N Clark St, Chicago, IL 60601 | POINT(-87.63098 41.88574) |      5
```

Reverse Geocoding

addy is an array of possible normalized addresses of a point location. You can expand aliases with unnest.

```
SELECT pprint_addy(unnest(rc.addy))
 FROM reverse_geocode(ST_Point(-87.63098, 41.88574) ) As rc;
```

If you are running PostGIS < 2.1.4, you will be missing the directional **N**.

```
           pprint_addy
--------------------------------
 102 N Lake St, Chicago, IL 60661
 200 N Clark St, Chicago, IL 60601
```

Links of Interest

- BUY OUR BOOKS HERE! http://www.postgis.us
- PostGIS
- Planet PostGIS

Promo Codes

44% off PostGIS In Action 2nd Edition **postgrescft**