

Greenplum Training

Labs

Greenplum PostGis

|  |  |
| --- | --- |
| : |  |
| : |  |
| : |  |

**Table of Contents**

[**1**](#_heading=h.gjdgxs) **Greenplum PostGis 3**

[1.1](#_heading=h.30j0zll) Objective 3

[1.2](#_heading=h.3znysh7) Steps to intiliaze the PostGis environment 3

[1.3](#_heading=h.2et92p0) Geometry queries 6

[1.4](#_heading=h.tyjcwt) Geometry relationships 8

[1.5](#_heading=h.3dy6vkm) USE CASE EXAMPLES - Geography 10

# Greenplum PostGis

## Objective

In this lab, you will test PostGis inside Greenplum.

## Steps to intiliaze the PostGis environment

1. Get the data <http://files.boundlessgeo.com/workshopmaterials/postgis-workshop-201401.zip>
2. Check if PostGis is installed

gppkg --query --all

You should see something like this:

20171013:04:04:27:006988 gppkg:gpdb-sandbox:gpadmin-[INFO]:-Starting gppkg with args: -q --all

madlib-ossv1.12\_pv1.9.9\_gpdb5.0

postgis-2.1.5

1. If not yet installed, download PostGis from Pivotal Network (<https://network.pivotal.io/>) and install it

gppkg -i postgis-2.1.5-gp5-rhel6-x86\_64.gppkg

1. Enable PostGis into the Greenplum database

$GPHOME/share/postgresql/contrib/postgis-2.1/postgis\_manager.sh mydatabase install

1. Check the version of PostGis

select postgis\_full\_version();

1. Obtain a table definition for a given shape file (extenuation .shp) by using the command

shp2pgsql -p filename.shp tablename

Obtain the table definition for the shapes below and create the DDL of the Greenplum table:

* nyc\_census\_blocks
* nyc\_homicides
* nyc\_neighborhoods
* nyc\_streets
* nyc\_subway\_stations

shp2pgsql -p nyc\_streets.shp nyc\_streets

Note: -p indicates “prepare”

You should see something like this:

Shapefile type: Arc

Postgis type: MULTILINESTRING[2]

SET CLIENT\_ENCODING TO UTF8;

SET STANDARD\_CONFORMING\_STRINGS TO ON;

BEGIN;

CREATE TABLE "nyc\_streets" (gid serial,

"id" float8,

"name" varchar(200),

"oneway" varchar(10),

"type" varchar(50));

ALTER TABLE "nyc\_streets" ADD PRIMARY KEY (gid);

SELECT AddGeometryColumn('','nyc\_streets','geom','0','MULTILINESTRING',2);

COMMIT;

The DDL of the Greenplum table will be:

drop table if exists nyc\_streets;

create table nyc\_streets

(

--gid serial, -- The field gid is not required

id text,

name text,

oneway text,

type text,

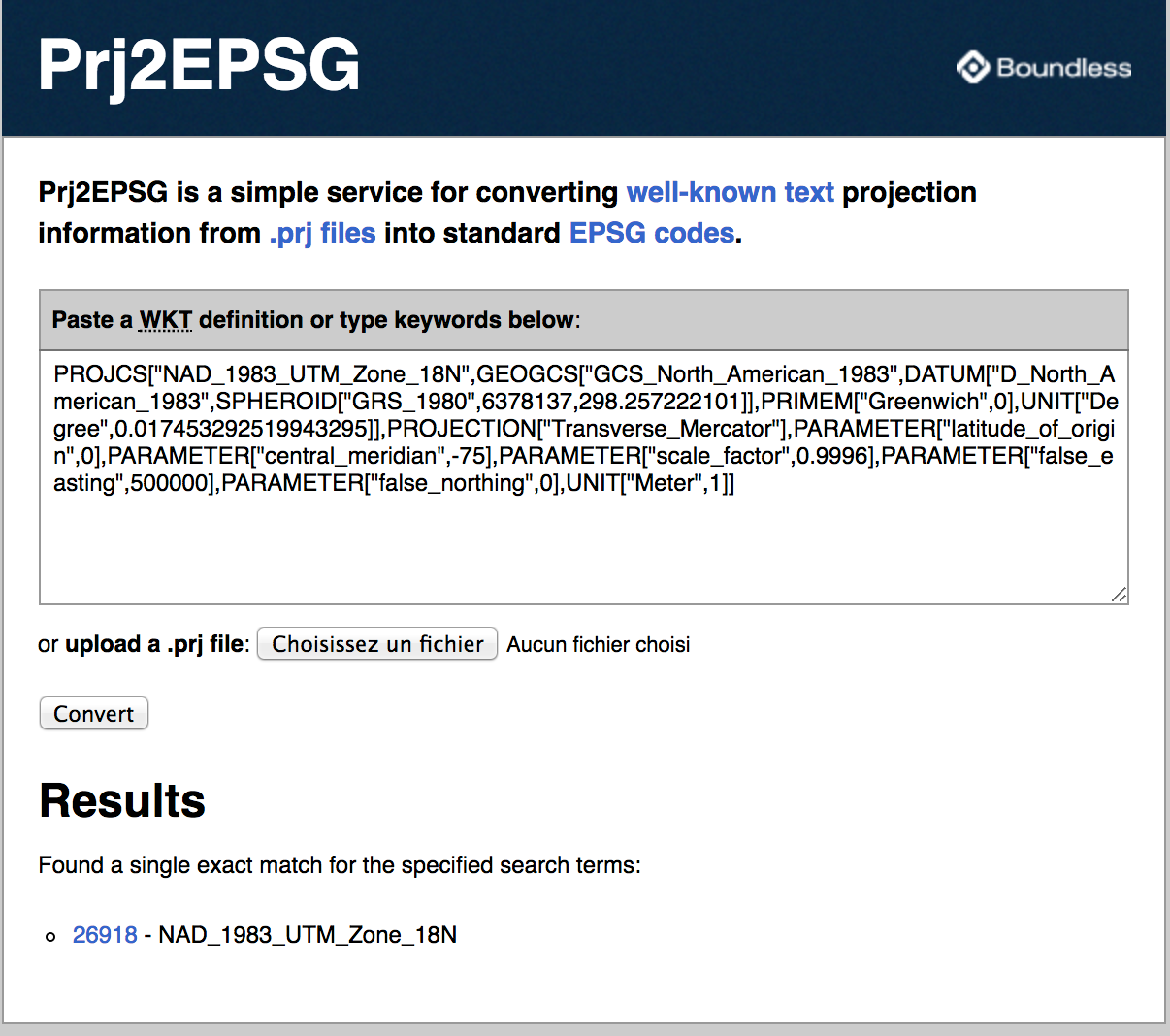
geom geometry

)

distributed randomly;

Note: The “AddGeometryColumn” command is directly defined as a new column in the GPDB DDL with type ‘geometry’.

1. Check projection files (extension .prj) from demo: <http://prj2epsg.org/search> to ensure correct SRID is set on loading with command line option -s



For the NYC Demographic demo, we will use **26918** as the SRID.

1. Use shp2pgsql to create a file of INSERT statements.

WARNING: This is slowest way to ingest spatial data into GPDB (perhaps as slow as a few 100 inserts per second) because it uses a discrete INSERT statement for each row executed through the mdw. That said, this method is simple, reliable, and suitable for small scale testing.

shp2pgsql command line options:

**-a (append)** writes the output file as INSERT statements that append rows to an existing table. In other words, it doesn't truncate and overwrite an existing table. This requires a table to be created prior to loading.

**-e (execute as a single transaction)** omits ‘BEGIN;’ and ‘COMMIT;’ lines in the output, which allows GPDB to commit each insert as a single transaction by default. this eliminates problems that could occur with transaction buffer limits

**-s nnnn** defines the SRID for the data.

**-D** creates a different format of data loading (using COPY) and much faster, so its recommended to use it.

shp2pgsql -a -D -s 26918 data/nyc\_streets nyc\_streets > nyc\_streets\_load.sql

shp2pgsql -a -D -s 26918 data/nyc\_census\_blocks nyc\_census\_blocks > nyc\_census\_blocks\_load.sql

shp2pgsql -a -D -s 26918 data/nyc\_homicides nyc\_homicides > nyc\_homicides\_load.sql

shp2pgsql -a -D -s 26918 data/nyc\_neighborhoods nyc\_neighborhoods > nyc\_neighborhoods\_load.sql

shp2pgsql -a -D -s 26918 data/nyc\_subway\_stations nyc\_subway\_stations > nyc\_subway\_stations\_load.sql

1. Load into Greenplum

psql -d tutorial -f nyc\_streets\_load.sql > log/nyc\_streets\_load.log 2>&1&

psql -d tutorial -f nyc\_census\_blocks\_load.sql > log/nyc\_census\_blocks\_load.log 2>&1&

psql -d tutorial -f nyc\_homicides\_load.sql > log/nyc\_homicides\_load.log 2>&1&

psql -d tutorial -f nyc\_neighborhoods\_load.sql > log/nyc\_neighborhoods\_load.log 2>&1&

psql -d tutorial -f nyc\_subway\_stations\_load.sql > log/nyc\_subway\_stations\_load.log 2>&1&

1. Have a look at the schema of all loaded tables

\dt

\d+ nyc\_streets

1. Verify the coordinate system of the data

select \* from geometry\_columns;

1. Find out the geometry data type of the geom value in each table, i.e. is it a POINT, LINE

NEIGHBORHOODS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CENSUS BLOCKS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HOMICIDES: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

STREETS: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SUBWAY STATIONS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

select ST\_AsText(geom)

,ST\_GeometryType(geom)

from nyc\_streets limit 1;

## Geometry queries

1. Find 3 largest neighborhoods by area:

Name neighborhood Area

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

select name, st\_area(geom)

from nyc\_neighborhoods

order by 2 desc

limit 3 ;

1. Find smallest boro by area: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

select boroname, sum(st\_area(geom))

from nyc\_neighborhoods

group by boroname

order by 2 asc

limit 1 ;

1. Find the smallest neighborhood in Manhattan by area: \_\_\_\_\_\_\_\_\_\_\_

select name, st\_area(geom)

from nyc\_neighborhoods

where boroname = 'Manhattan'

order by 2 asc

limit 1 ;

1. What’s the area of Manhattan in ha and in km²?

select sum(st\_area(geom)) / 10000 as ha

, sum(st\_area(geom)) / 1000000 as km2

from nyc\_neighborhoods

where boroname = 'Manhattan';

1. Locate latitude and longitude of the Wall street station on 2&3 red line in Manhattan: \_\_\_\_\_\_\_\_\_\_\_\_\_

select long\_name,routes,color

,ST\_AsText(ST\_Transform(ST\_SetSRID(geom,26918),4326))

from nyc\_subway\_stations

where name = 'Wall St'

and routes = '2,3'

and color = 'RED';

1. Calculate number of miles (as the crow flies) from Wall street to Times Square subway stations on the 2&3 red line: \_\_\_\_\_\_\_\_\_\_\_\_\_

select st\_distance(Wall\_street\_geom,Times\_Square\_geom) / 1609.34 as distance\_miles

from (select max(geom) FILTER (WHERE name ='Wall St') as Wall\_street\_geom

, max(geom) FILTER (WHERE name ='Times Sq') as Times\_Square\_geom

from nyc\_subway\_stations

where name in ('Wall St','Times Sq')

and color = 'RED') T;

1. All Atlantic Avenue (Brooklyn) subway stations are Closed. Find 3 nearest stations and what is their distance? \_\_\_\_\_\_\_\_\_\_\_\_\_

select T\_Atlantic.long\_name

,T\_other.borough

,T\_other.long\_name

,T\_other.routes

,T\_other.color

,st\_distance(T\_other.geom,T\_Atlantic.geom) as distance

from

(select \*

from nyc\_subway\_stations

where not (borough = 'Brooklyn'

and long\_name ilike 'Atlantic Ave%')) T\_other

, (select \*

from nyc\_subway\_stations

where borough = 'Brooklyn'

and long\_name ilike 'Atlantic Ave%') T\_Atlantic

order by distance asc

limit 3

1. Locate the 3 closest subway stations to LaGuardia Airport in order of closeness: \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

select borough

,long\_name

,routes

,color

,st\_distance(Geography(ST\_Transform(geom,4326)),

ST\_GeographyFromText('POINT(-73.872013 40.774490)')) -- LaGuardia Longitude / Latitude

/ 1000

as distance

from nyc\_subway\_stations

order by distance asc

limit 3

1. What’s the total length of streets in NYC in km?

select sum(st\_length(geom)) / 1000 as km

from nyc\_streets;

1. How long is ‘Columbus Cir’?

select st\_length(geom)

from nyc\_streets

where name = 'Columbus Cir';

1. What is the length of streets in NYC, by types?

select type, sum(st\_length(geom)) as meters

from nyc\_streets

group by type

order by 2;

## Geometry relationships

1. Atlantic commons belongs to which neighborhood and which borough?

select name, boroname

from nyc\_neighborhoods as a,

(

select geom as ls

from nyc\_streets

where name='Atlantic Commons'

) as b

where st\_intersects(a.geom, b.ls);

1. How many people live on Atlantic Commons? (within 50 meters)

select

sum(popn\_total::integer) as "total population"

from

nyc\_census\_blocks as a,

(

select geom

from nyc\_streets

where name='Atlantic Commons'

) as b

where

st\_dwithin(a.geom, b.geom, 50);

1. What subway station is in ‘Little Italy’? what subway route is on it?

select

s.name,

s.routes

from

nyc\_subway\_stations as s,

nyc\_neighborhoods as n

where

st\_contains(n.geom, s.geom)

and n.name = 'Little Italy';

1. What are neighborhoods served by 6-train?

select

n.name,

n.boroname

from

nyc\_subway\_stations as s,

nyc\_neighborhoods as n

where

st\_contains(n.geom, s.geom)

and s.routes = '6'

group by 1,2

order by 2;

1. How many people live in neighborhood ‘Battery Park’?

select

sum(popn\_total::integer) as population

from

nyc\_neighborhoods as n,

nyc\_census\_blocks as c

where

st\_intersects(n.geom, c.geom)

and n.name = 'Battery Park';

1. What are the population density (persons/sq km) of the ‘Upper West Side’ and ‘Upper East Side’?

select

n.name,

sum(c.popn\_total::integer)/(st\_area(n.geom)/1000000) as density

from

nyc\_census\_blocks as c,

nyc\_neighborhoods as n

where

st\_intersects(c.geom, n.geom)

and (n.name = 'Upper West Side' or n.name='Upper East Side')

group by 1, n.geom;

## Geography vs geometry

1. Calculate distance from Los Angeles to Paris

SELECT ST\_Distance(

ST\_GeographyFromText('POINT(-118.4079 33.9434)'), -- Los Angeles (LAX)

ST\_GeographyFromText('POINT(2.5559 49.0083)') -- Paris (CDG)

);

1. Compare geometrical and geographic distance

select

st\_distance(

st\_geometryfromtext('Point(-118.4079 33.9434)', 4326), -- Los Angeles (LAX)

st\_geometryfromtext('Point(2.5559 49.0083)', 4326) -- Paris (CDG)

) as geometry\_distance,

st\_distance(

st\_geographyfromtext('Point(-118.4079 33.9434)'), -- Los Angeles (LAX)

st\_geographyfromtext('Point(2.5559 49.0083)') -- Paris (CDG)

) as geography\_distance;

Note: The units of measurement for geometry with SRID=4326 is in units “degrees”. So the distance from LosAngeles to Paris is 121 degrees.

geography distance is simply meters -- so the answer is 9124 Km….

1. Calculate distance between several cities

You can use the site <http://www.geobytes.com/citydistancetool/> in order to get longitude and latitude + the result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Target City | Latitude | Longitude | Distance to Reston, VA (km) | Distance to Reston, VA (degree) |
| Reston, VA |  |  |  |  |
| Helsinki, Finland |  |  |  |  |
| Dublin, Ireland |  |  |  |  |
| Portland, Maine |  |  |  |  |
| Philadelphia, PA |  |  |  |  |
| Baltimore, MD |  |  |  |  |

select

target\_order

,'Reston' as orig\_city

,target\_city

,st\_distance(ST\_GeographyFromText(target\_coord),ST\_GeographyFromText('POINT(-77.344200 38.962299)'))/1000 as distance\_km

,st\_distance(ST\_GeometryFromText(target\_coord) ,ST\_GeometryFromText('POINT(-77.344200 38.962299)')) as distance\_degree

from (

select 1 as target\_order, 'Reston' as target\_city,'POINT(-77.344200 38.962299)' as target\_coord

union

select 2 as target\_order, 'Helsinki','POINT(24.934000 60.175999)'

union

select 3 as target\_order, 'Dublin','POINT(-6.250000 53.333000)'

union

select 4 as target\_order, 'Portland','POINT(-70.261101 43.669998)'

union

select 5 as target\_order, 'Philadelphia','POINT(-75.216003 39.930199)'

union

select 6 as target\_order, 'Baltimore','POINT(-76.638397 39.289398)'

) T