

PostGIS Short Course (10 Slides, 20 Minutes)

Jeff D. Hamann

Forest Informatics
PO Box 1421
Corvallis, Oregon 97339-1421

August 27, 2010

Every GIS needs a system to handle:

- ① geometries (points, lines, polygons)
- ② spatial reference system (SCS+Datum+Ellipsoid)
- ③ spatial functionality (measurements, operators)

PostGIS handles:

- ① geometries using `geometry_columns` and `geos 3.1.1`
- ② spatial reference systems using `spatial_ref_sys` and `proj.4`
- ③ spatial functionality using over 300 sql functions
- ④ and a whole lot more!

Step 1 - Installing Dependencies (zzz)

You will need to install the following libs

- ① GEOS (Geometry Engine - Open Source) includes all the OpenGIS Simple Features for SQL spatial predicate functions and spatial operators, as well as enhanced topology functions.
- ② PROJ.4 - Cartographic Projections Library originally written by Gerald Evenden then of the USGS.
- ③ GDAL/OGR is a translator library for raster/vector geospatial data formats. It also comes with a variety of useful command-line utilities for data translation and processing.

```
$ tar xvfz geos-3.1.1.tar.bz2
$ cd geos-3.1.1
$ ./configure; make; make check; make install; cd ..
```

Creating a PostGIS database

```
psql -d yourdatabase -f postgis.sql  
psql -d yourdatabase -f spatial_ref_sys.sql  
psql -d yourdatabase -f postgis_comments.sql
```

```
$ psql -h localhost -U postgres yourdatabase  
psql (8.4.4)  
Type "help" for help.
```

```
yourdatabase=# select version();  
yourdatabase=# select postgis_version();  
yourdatabase=# select * from PostGIS_Full_Version();
```

Importing Data

From scratch using SQL (pgissc.sql):

```
insert into boundary (bid,area_units,bdry) values \  
( 1, -1.0, GeometryFromText( 'MULTIPOLYGON((( \  
55.000000 45.000000, 80.000000 45.000000, 80.000000 \  
65.000000, 55.000000 65.000000, 55.000000 45.000000)))', \  
4326 ) );
```

Using ogr2ogr For Personal Geodatabases (MDB files):

```
C:\>ogr2ogr -overwrite \  
-f "PostgreSQL" \  
PG:"host=192.168.0.7 user=hamannj dbname=tachrv12" \  
-nln user_roads2 -a_srs EPSG:2286 BNDRY.mdb \  
layer Roads_24K_WA
```

ogr2ogr process for output is same as input arguments, only backwards.

PGISSC Example

- 1 create data and ingest forest data using `pgissc.sql`
- 2 generate a boundary and an adjacency table using `pgsql`
- 3 generate views in `pgissc.sql`
- 4 generate kml feed from view using `ogr2ogr (pgissc.kml)`
- 5 open `pgissc.kml`

geometry_columns

```
pgisss=# \d+ geometry_columns
```

Table "public.geometry_columns"

Column	Type	Modifiers	Storage	Description
f_table_catalog	character varying(256)	not null	extended	
f_table_schema	character varying(256)	not null	extended	
f_table_name	character varying(256)	not null	extended	
f_geometry_column	character varying(256)	not null	extended	
coord_dimension	integer	not null	plain	
srid	integer	not null	plain	
type	character varying(30)	not null	extended	

Indexes:

"geometry_columns_pk" PRIMARY KEY, btree (f_table_catalog, f_table_schema, f_table_name, f_geometry_column)

Has OIDs: yes

```
pgisss=#
```

```
pgisss=# select * from geometry_columns;
```

f_table_catalog	f_table_schema	f_table_name	f_geometry_column	coord_dimension	srid	type
	public	stands	boundary	2	4326	MULTIPOLYGON
	public	stands	location	2	4326	POINT

(2 rows)

```
pgisss=#
```

```
pgisss=# \d+ spatial_ref_sys
```

Table "public.spatial_ref_sys"				
Column	Type	Modifiers	Storage	Description
srid	integer	not null	plain	
auth_name	character varying(256)		extended	
auth_srid	integer		plain	
srtext	character varying(2048)		extended	
proj4text	character varying(2048)		extended	

Indexes:

"spatial_ref_sys_pkey" PRIMARY KEY, btree (srid)

Has OIDs: no

```
pgisss=# select * from spatial_ref_sys limit 1;
```

```
-[ RECORD 1 ]-----  
srid          | 3819  
auth_name     | EPSG  
auth_srid     | 3819  
srtext        | GEOGCS["HD1909",DATUM["Hungarian_Datum_1909",SPHEROID["Bessel 1841",6377397.155,299.1528128,A  
proj4text     | +proj=longlat +ellps=bessel +towgs84=595.48,121.69,515.35,4.115,-2.9383,0.853,-3.408 +no_defs
```

```
pgisss=#
```


Data Types and Functions

- 1 Geometry creation, [E]WKT/[E]WKB, accessors, operators, aggregates, editors, AS_asKML, ST_asXYZ, and bbox type A && B)
- 2 Functions **almost** always start with ST_
- 3 Go to www.bostongis.com
- 4 Get the cheatsheet. It's got everything!
- 5 `$ogr2ogr ---formats`