PostGIS Short Course (10 Slides, 20 Minutes)

Jeff D. Hamann

August 27, 2010

1 Preface

Our PostGIS training class teaches you the fundamentals of Geographic Information Systems (GIS) including common conventions and methods of managing and manipulating geospatial data using the open-source PostGIS extensions to PostgreSQL.

Students attending this class will learn how to install, query, import, export, manage, and manipulate geospatial data within the PostgreSQL/PostGIS environment.

Students attending this class should have the following:

1. An understanding of relational database fundamentals, including the ability to write SQL queries.

2. The ability to perform basic file management and editing tasks using the Linux file system (GUI editors are fine).

2 Introduction (09:00-09:15)

A geographic information system (GIS), or geographical information system, is any system that captures, stores, analyzes, manages, and presents data that are linked to location. In the simplest terms, GIS is the merging of cartography and database technology. GIS systems are used in cartography, remote sensing, land surveying, photogrammetry, geography, urban planning, emergency management, navigation, and localized search engines.

3 GIS Fundamentals

This is all text that happens in the article only.

3.1 Geometries and Data Types (09:30-09:45)

3.1.1 Vector data (09:35-09:40)(5 min)

In a GIS, geographical features are often expressed as vectors, by considering those features as geometrical shapes. Different geographical features are expressed by different types of geometry:

Points A simple vector map, using each of the vector elements: points for wells, lines for rivers, and a polygon for the lake.

Zero-dimensional points are used for geographical features that can best be expressed by a single point reference; in other words, simple location. For example, the locations of wells, peak elevations, features of interest or trailheads. Points convey the least amount of information of these file types. Points can also be used to represent areas when displayed at a small scale. For example, cities on a map of the world would be represented by points rather than polygons. No measurements are possible with point features.

Lines or polylines One-dimensional lines or polylines are used for linear features such as rivers, roads, railroads, trails, and topographic lines. Again, as with point features, linear features displayed at a small scale will be represented as linear features rather than as a polygon. Line features can measure distance.

Polygons Two-dimensional polygons are used for geographical features that cover a particular area of the earth's surface. Such features may include lakes, park boundaries, buildings, city boundaries, or land uses. Polygons convey the most amount of information of the file types. Polygon features can measure perimeter and area.

3.1.2 Raster data (09:30-09:35)(5 min)

A raster data type is, in essence, any type of digital image represented in grids. Anyone who is familiar with digital photography will recognize the pixel as the smallest individual unit of an image. A combination of these pixels will create an image, distinct from the commonly used scalable vector graphics which are the basis of the vector model. While a digital image is concerned with the output as representation of reality, in a photograph or art transferred to computer, the raster data type will reflect an abstraction of reality. Aerial photos are one commonly used form of raster data, with only one purpose, to display a detailed image on a map or for the purposes of digitization. Other raster data sets will contain information regarding elevation, a digital elevation model, or reflectance of a particular wavelength of light, Landsat.

Raster data type consists of rows and columns of cells, with each cell storing a single value. Raster data can be images (raster images) with each pixel (or cell) containing a color value. Additional values recorded for each cell may be a discrete value, such as land use, a continuous value, such as temperature, or a null value if no data is available. While a raster cell stores a single value, it can be extended by using raster bands to represent RGB (red, green, blue) colors, colormaps (a mapping between a thematic code and RGB value), or an extended attribute table with one row for each unique cell value. The resolution of the raster data set is its cell width in ground units.

Raster data is stored in various formats; from a standard file-based structure of TIF, JPEG, etc. to binary large object (BLOB) data stored directly in a relational database management system (RDBMS) similar to other vector-based feature classes. Database storage, when properly indexed, typically allows for quicker retrieval of the raster data but can require storage of millions of significantly-sized records.

3.2 Spatial Reference System

A spatial reference system (SRS) is a system of three components to store and maintain geospatial data using a combination attributes that defines: 1) an ellipsoid, spheroid, or geoid 2) a datum using that ellipsoid or spheroid or geoid, and 3) a spatial coordinate system that uses one of the following: a geocentric, geographic, or projection coordinate system (projection) where the projection has a geographic coordinate system associated with it.

3.3 Spatial Functions

This is the text that will go into the spatial functions section of the article/presentation.

4 Introduction to PostGIS

PostGIS is an open source software program that adds support for geospatial data to the PostgreSQL object-relational database.

PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium.

As such, PostGIS includes:

- Geometry types for points, linestrings, polygons, multipoints, multilinestrings, multipolygons and geometrycollections.
- Spatial predicates for determining the interactions of geometries using the 3x3 Egenhofer matrix (provided by the GEOS software library).
- Spatial operators for determining geospatial measurements like area, distance, length and perimeter.
- Spatial operators for determining geospatial set operations, like union, difference, symmetric difference and buffers (provided by GEOS).
- R-tree-over-GiST (Generalised Search Tree) spatial indexes for high speed spatial querying.
- Index selectivity support, to provide high performance query plans for mixed spatial/non-spatial queries.
- For raster data, under development PostGIS WKT Raster (we won't talk about this)

The PostGIS implementation is based on "light-weight" geometries and indexes optimized to reduce disk and memory footprint. Using light-weight geometries helps servers increase the amount of data migrated up from physical disk storage into RAM, improving query performance substantially.

The first version was released in 2001 by Refractions Research under the GNU General Public License. A stable "1.0" version was released on April 19, 2005, which followed 6 release candidates. In 2006, PostGIS was certified as a compliant Simple Features for SQL database by the Open Geospatial Consortium. That's a big deal because it

4.1 Installing PostGIS

Installing PostGIS is relatively simple, but it does require some steps before we can compile the library itself.

4.2 Creating a PostGIS Enabled Database

This is the text that will go into the article.

This will go into the article only.

```
rufus=# select postgis_version();
            postgis_version
---------------------------------------
 1.5 USE_GEOS=1 USE_PROJ=1 USE_STATS=1
(1 row)

rufus=# select postgis_full_version();
                                          postgis_full_version
-------------------------------------------------------------------------------------------------------
 POSTGIS="1.5.1" GEOS="3.1.1-CAPI-1.6.0" PROJ="Rel. 4.7.1, 23 September 2009" LIBXML="2.7.3" USE_STATS
(1 row)

rufus=#
```

4.3 Importing/Exporting Data

Basic data import/export functions using external tools (OGR, etc.)

Since you most likely will be importing data from some other source, you will want to import existing data.

The major tool for importing data into and out of a PostGIS enabled database is the OGR tools provided with the GDAL library.

5 An Example

Present `pgissc.kml` example.

6 In Depth PostGIS Tables and Functions

6.1 geometry_columns

6.2 spatial_ref_sys

If you do a query on the table

6.3 Functions

To perform our analysis, we need to codify our objectives functions, constraints, and accounting variables.

In this section, we present some functions that are commonly used.

2.3) A few slides on what functions are available, by type and purpose.

We assume that any function you need can be expressed as some sequence of function calls to PostIGIS, PostgreSQL, or R functions possible.

Here, we define a simple query using the data we generated by hand in Section

6.4 Results

this is what is supposed to go into the results section.