PostGIS

Info Sheet

Geospatial Extensions for PostgreSQL :

Overview

PostGIS is a free, open source spatial extender for the PostgreSQL open source database. Spatial extenders leverage existing database capabilities to handle geographical and image data. Common applications include proximity analysis for addressing questions of how far something is, geocoding for finding latitude and longitude of addresses, and web mapping for supplying data to various map servers.

Basic Vector Data Types

The geometry types supported by PostGIS include vector geometries:

- Points / Multi-Points
- Lines / Multi-Lines
- Polygons / Multi-Polygons

These geometric data elements are encoded as *geometries* or *geographies*.

Geometry is the planar vector type, where the spatial reference determines the underlying coordinate system as well as the unit of measure.

Geography is the *Geodetic* type, where the units of measurement are always in meters and coordinates in degrees.

Advanced Geospatial Data Types

Raster is a pixel-based, multi-band type similar to common image types. The raster may be georeferenced by having a set spatial reference (SRID).

TopoGeometry is a relational vector type that represents a network of connected nodes (points), edges (linestrings), and faces (polygons). The topogeometry is made up of these elemental components and is referred to as punctal (if point), lineal (if linestring/multilinestring), or areal (if polygon/ multipolygon).

Geomval is a compound type that arises when you intersect a geometry with a raster - for example, if you overlay a polygon or linestring atop a single-band raster.

Online Cheat sheets and Reference Cards

- General: http://www.postgis.us/downloads/postgis21_cheatsheet.pdf
- Raster: http://www.postgis.us/downloads/postgis21_raster_cheatsheet.pdf
- Topology: http://www.postgis.us/downloads/postgis21_topology_cheatsheet.pdf

Recipes:

Join two tables based on spatial intersection

In:
```
SELECT a.col_1, b.col_2
FROM geoTableA a
JOIN geoTableB b
  ON (a.polygonCol_z && b.pointCol_y)
WHERE
 ST_INTERSECTS(a.polygonCol_z,b.pointCol_y)
```

Aggregating Data Values within Spatial Areas

In:
```
SELECT a.col_1, AVG(b.valcol_2)
FROM geoTableA a
JOIN geoTableB b
  ON (a.polygonCol_z && b.pointCol_y)
WHERE
 ST_INTERSECTS(a.polygonCol_z,b.pointCol_y)
```

Creating a table with a spatial column

In:
```
-- Create a basic PostgreSQL table
CREATE TABLE geoTableB (id serial);
-- Add a spatial column to the table
--   SRID 4326, 2 Dimensional Point
SELECT AddGeometryColumn('my_schema_name',
     'geoTableB','geom',4326,'POINT',2);
```

Adding a Spatial Index

In:
```
CREATE INDEX idx_geoTableB_points
 ON geoTableB USING GIST(pointCol_y)
```

Read more about spatial indexes:
https://postgis.net/docs/using_postgis_dbmanagement.html#idm2267

More Samples:
https://postgis.net/docs/using_postgis_dbmanagement.html#examples_spatial_sql

Other References and Links:

PostgreSQL Cheatsheet

http://www.postgresqltutorial.com/postgresql-cheat-sheet/

GIS Stack Exchange – PostGIS

https://gis.stackexchange.com/questions/tagged/postgis

Spatial Indexing

http://workshops.boundlessgeo.com/postgis-intro/indexing.html