Spatial Databases

An Introduction to PostGIS

Matthew Wigginton Conway

July 25, 2014

(slides available at `http://www.indicatrix.org/spdb.pdf`)

Spatial Databases

- Allow us to ask spatial questions of a database
- e.g. How far is the nearest fire station from every building in Chicago?
- or how many parcels are in each Census tract in Des Moines?

Types of Spatial Data

- Raster (recorded on a regular lattice)
- Vector
  - Point
  - Line
  - Polygon
- Vector data generally consists of the shapes and associated attribute information

Coordinate Systems and Projections

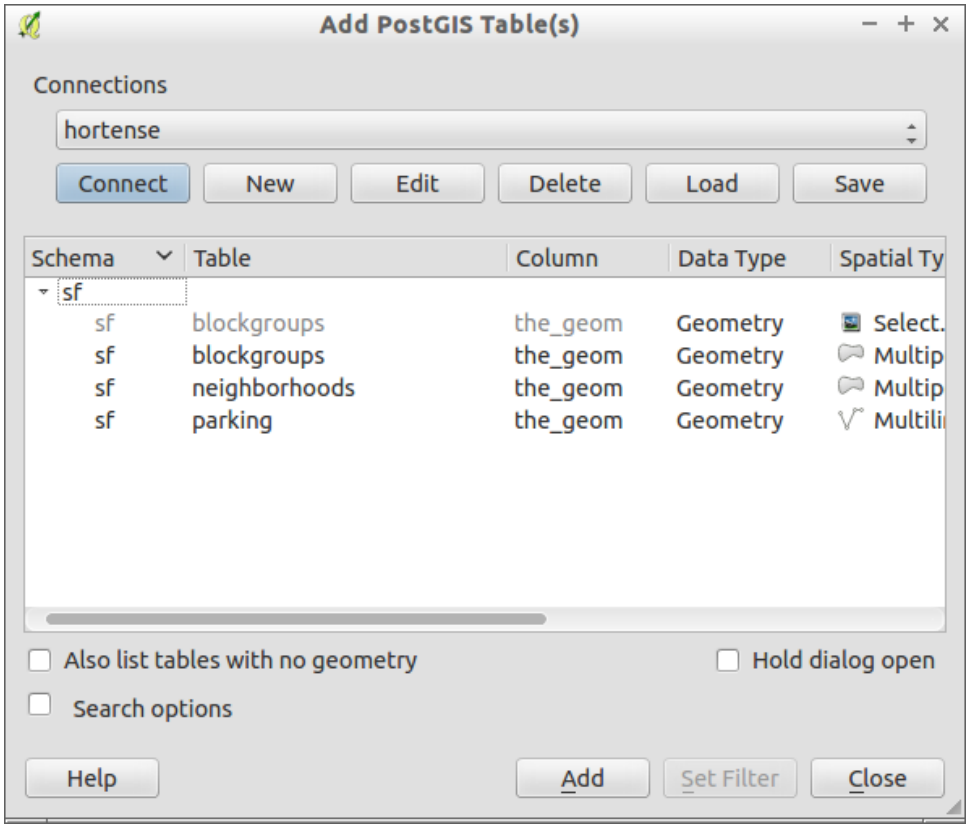
- Geographic coordinate systems
  - Represent positions as degrees north of the equator and east of the prime meridian
  - e.g. our office is 41.87856, -87.62727 in WGS 1984
  - Defined based on a ellipsoid which represents the shape of the Earth and a datum which matches that ellipsoid to the actual earth
  - Calculations require more complex math
- Projected coordinate systems
  - Represent positions on a plane, in distance units
  - e.g. our office is 1176532 ft E, 1899128.15793387 ft W in Illinois State Plane East, US Survey Feet
  - Allow simpler and faster math
  - Introduce distortion, which is particularly apparent in large regions

Spatial SQL

- PostGIS adds a number of functions to SQL for spatial manipulation
- Most (all?) begin with ST_
- For example, SELECT ST_Centroid(the_geom) FROM census_tracts; would return the centroid of the feature
- SELECT * FROM service311calls s LEFT JOIN census_tracts t ON (ST_Within(s.the_geom, t.the_geom)); would select attributes of 311 calls and the attributes of the census tract that contains them.
- In PostGIS, geographic data is just another column (of type Geometry(class, projection) or Geography).

Accessing PostGIS in Desktop GIS

QGIS can access PostGIS data directly by going to Layer ▷ Add PostGIS Layer:



Hands-on Demo

Open your favorite Postgres client
Database connection parameters:
(omitted)

Datasets

- All in the sf schema
- sf.parking: parking counts by block in San Francisco
- sf.neighborhoods: neighborhood boundaries in San Francisco
- sf.blockgroups: block-group population

Datasets

How many parking spaces are in each neighborhood?

```
SELECT n.name, SUM(prkng_sply)
FROM sf.parking p
LEFT JOIN sf.neighborhoods n ON
  (ST_Within(ST_Centroid(p.the_geom), n.the_geom))
GROUP BY n.name
ORDER BY 1;
```

How many people live in each neighborhood?

```
SELECT n.name, SUM(pop)
FROM sf.blockgroups b
LEFT JOIN sf.neighborhoods n ON
  (ST_Within(ST_Centroid(b.the_geom), n.the_geom))
GROUP BY n.name
ORDER BY 1;
```

Caveat emptor: a word on projections

- Data stored as latitude and longitude has limitations
  - When stored in latitude and longitude, the distance from Foothill College (Los Altos Hills, CA) to the University of Chicago is 34.8 degrees—a meaningless figure
- Latitudes and longitudes should be stored as geography
  - Not all operations work (yet) on geographies
  - Geographies are slower
  - But the distance from Foothill College to the University of Chicago is 2,986 km, the correct great-circle distance
- Data stored in different projections cannot be compared (PostGIS will yield an error)
- ST_Transform is your friend
- Local analyses can often use the State Plane Coordinate System or Universal Transverse Mercator

Caveat emptor: never store data in the public schema

Well, you can, but PostGIS also keeps track of its internal state in that schema. Separating your data into other schemas makes backup, restore, and PostGIS upgrades painless.

Questions and additional resources

- `http://postgis.net`: PostGIS web page and documentation
- `http://epsg.io`: Standard coordinate system descriptions
- `http://qgis.org`: Quantum GIS, open-source Geographic Information System with PostGIS support