# Spatial Database Management
# GEP 664 / GEP 380
## Class #10: Organizing spatial data

Frank Donnelly

Dept of EEGS, Lehman College CUNY

Spring 2019

## Today's Topics



## Studies

Examples of studies that employed spatial databases:

Baloye2016 Nigerian Disaster Management
Damito2018 US Archaeological Museum Collection
Favretto2018 Cataloging Ancient Roman Coins
Furnass2013 Water Quality in UK Pipe Networks
Oussalah2013 Twitter and Geolocation
Rosser2017 New Zealand Landslides
Silavi2016 Modeling Responsive Urban Environments
Tissot2012 French Oyster Farms
Vias2018 Hiking Routes in Spain

## Considerations

- What are the goals for my database? What topic do I want to investigate? What problem do I want to solve?
- What data do I need to model my topic? What data do I need to answer my questions?
- What kinds of analysis would I do?
- What data is available? How do I obtain or collect it? How do I process it?
- How do I construct my database to model my topic? What are the entities, attributes, and relationships? How does the data fit into the database model?
- How do I insure the integrity and efficiency of my data?
- How can I summarize and visualize my data?

## Scale and Generalization - Global

Spatial Considerations

Do boundaries need to be precise, or should they be generalized? Does the precision impact your analysis? What would be most appropriate for your maps?

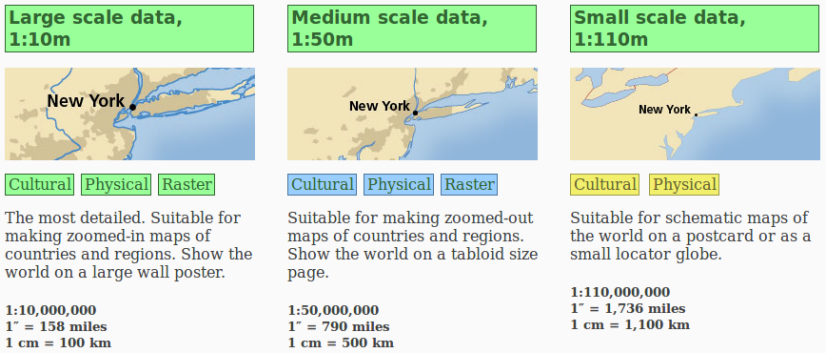



Image source: `http://www.naturalearthdata.com/downloads/`

## Scale and Generalization - Local

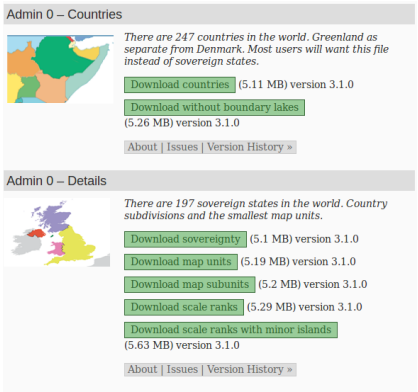
Spatial Considerations

Detailed

Generalized

## Definitions

Spatial Considerations




What is a country? Just independent states, or dependent states and territories? Legal or land boundaries? Polygons or lines?

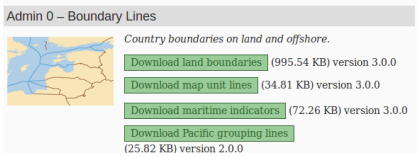



Image source: `http://www.naturalearthdata.com/downloads/`

## Summary of Spatial Considerations

- Scale
- Generalization
- Time
- Locational definitions
- Spatial Reference System
- Identifiers between layers and attributes

## Today's Topics

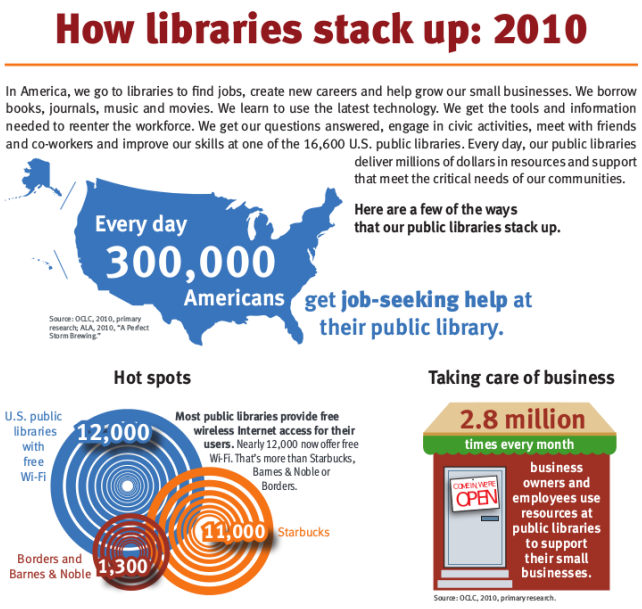


## Question

What is the geographic distribution of public libraries in the United States? Are some areas better served than others?

Why is this important? Public libraries play an important role in civic life. They provide a range of informational resources and services and a public space that benefits communities. Communities that have no public library or are distant from one will not realize these benefits.

## Research Inspiration



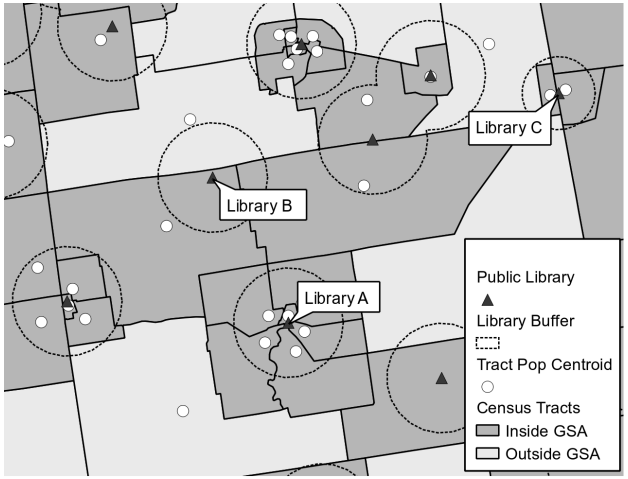

## Possible Methods for Study

Take local data (block groups, tracts, ZIP Codes) and summarize and analyze at a large level (counties, metropolitan areas, states, regions).

- Count the number of libraries in each community
- Create zones of impact (buffers) and count areas and people inside and outside the zones
  - Zones could be fixed-width or variable
  - Selected areas could be any that intersect, or that have their centroid in, or can be apportioned based on percentage of overlap
- Measure distance from community to nearest library and study variations in distance
  - Distance could be straight-line or network-based
  - Distance to nearest, n nearest, or avg to all
  - When summarized distance could be weighted by population

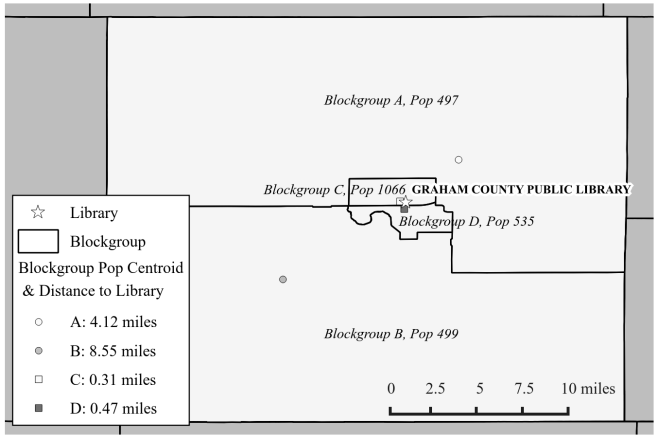
## Method 1 - Variable Buffers

Census tract counted as in library zone if: it has a library, or population centroid falls within buffer. Make comparisons between areas that are in versus out.




## Method 2 - Population-Weighted Distance

Distance calculated from census block group population centroids to nearest library. Average population-weighted distance calculated for all areas.




## Chosen Method

"Regional variations in average distance to public libraries in the United States" Donnelly LISR 37 (2015): 280-289.

- Population-weighted straight-line distance
- Population centroids of census blockgroups with 2010 pop to closest library
- Summarized by US Regions for metro and non-metro
- Summarized and mapped by state
- LISA for states to identify clusters
- Spearman correlations between state and library summaries:
  - Population density and concentration (Hoover Index)
  - % of population that is urban and metropolitan

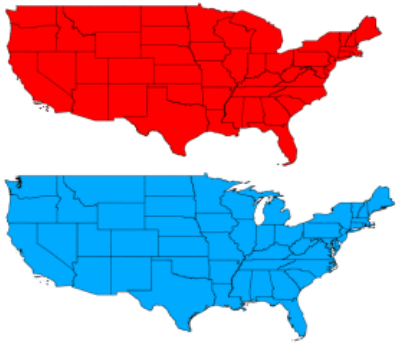
## What data do I need?

- Locations of all public libraries (IMLS)
  - Drop bookmobiles and US territories
  - Geocode missing coordinates
  - Build geometry from coordinates
- Census block group centroids (includes 2010 pop)
  - Build geometry from coordinates
- Boundaries and attributes for context and creating summaries
  - States - for summaries by state and division, attributes for density and urbanity
  - Counties - for summaries by metro areas, needed for calculating state metro population

## Projection, Generalization, and Scale

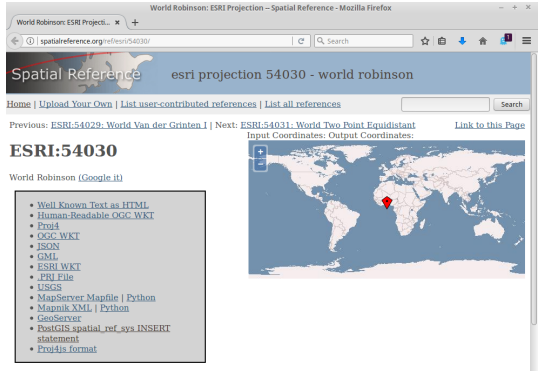
All layers and coordinates are in NAD 83. Libraries and centroids span the continent, but measured distances will all be short. Add the SRS for North America Lambert Conformal Conic to the PostGIS spatial ref system table and transform.

Distances will be between points which are relatively precise. Boundaries are just for visual depiction. Use Census Generalized Cartographic Boundary Files instead of TIGER.



## Expanding the SRS Table

The spatial_ref_sys table omits several continental and global map projections that are not defined by EPSG. Definitions and PostGIS Insert Statements for many SRS are available at Spatial Reference: `http://spatialreference.org/`

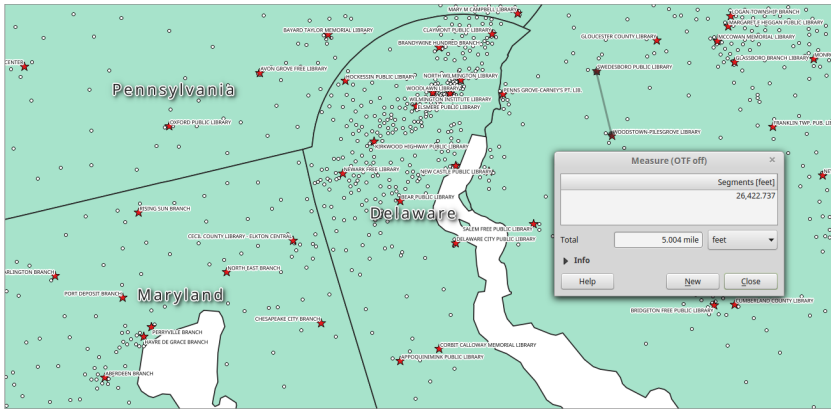



## Loading

- Load and plot coordinates of libraries and bg pop centroids, build with ST_Point and transform to NA LCC
- Load county attribute table that indicates metro status
- Load state and block groups shapefiles with PostGIS, create new tables and transform to NA LCC
- Gather and load attributes for the states (density, urbanity, etc)
- The GEOID of the bg pop centroids contains: state, county, tract, and bg codes. Split into separate columns.
- ADD COLUMN then UPDATE - SET - FROM to add attributes to the bg pop centroids:
  - Insert the division code from the state file
  - Insert metro status from the county file

## Ready to Go

Libraries (red stars), BG Population Centroids (white dots)



## Calculate Distances

- Use ST_Distance and DISTINCT ON to calculate the distance from every bg centroid to the closest library, save this data in a new table.

TABLE dist_blkgrps Search Show All

| rowid | InputID | TargetID | Distance |
|---|---|---|---|
| 1 | 010010201001 | AL0192-002 | 1782.52524632 |
| 2 | 010010201002 | AL0192-002 | 3128.08402211 |
| 3 | 010010202001 | AL0192-002 | 2284.43882012 |
| 4 | 010010202002 | AL0192-002 | 1013.49701842 |
| 5 | 010010203001 | AL0192-002 | 2229.24571565 |
| 6 | 010010203002 | AL0192-002 | 1342.15900589 |
| 7 | 010010204001 | AL0192-002 | 2932.38932161 |
| 8 | 010010204002 | AL0192-002 | 2900.73693274 |
| 9 | 010010204003 | AL0192-002 | 2310.24541416 |
| 10 | 010010204004 | AL0192-002 | 2318.02883892 |

## Gather Data Together

- Create another table where the distance column is associated with the block groups and all geographic identifiers. Convert distance to miles.

| geoid2 | dist | pop2010 | statefp | countyfp | tractce | blkgrpce | usps | region | metro |
|---|---|---|---|---|---|---|---|---|---|
| 010010201001 | 1.107609494... | 698 | 01 | 001 | 020100 | 1 | AL | 3 | M |
| 010010201002 | 1.943700696... | 1214 | 01 | 001 | 020100 | 2 | AL | 3 | M |
| 010010202001 | 1.419484034... | 1003 | 01 | 001 | 020200 | 1 | AL | 3 | M |
| 010010202002 | 0.629757655... | 1167 | 01 | 001 | 020200 | 2 | AL | 3 | M |
| 010010203001 | 1.385188639... | 2549 | 01 | 001 | 020300 | 1 | AL | 3 | M |
| 010010203002 | 0.833978683... | 824 | 01 | 001 | 020300 | 2 | AL | 3 | M |
| 010010204001 | 1.822101685... | 944 | 01 | 001 | 020400 | 1 | AL | 3 | M |
| 010010204002 | 1.802433808... | 1937 | 01 | 001 | 020400 | 2 | AL | 3 | M |
| 010010204003 | 1.435519503... | 935 | 01 | 001 | 020400 | 3 | AL | 3 | M |
| 010010204004 | 1.440355897... | 570 | 01 | 001 | 020400 | 4 | AL | 3 | M |

## Calculate Weighted Distances for Larger Areas

Create various Views with GROUP BY statement. Calculate the average population-weighted distance (US regions by metro area below)

```
SELECT region, metro,
ROUND((SUM(dist*pop2010))/SUM(pop2010),1) AS wgt_dist
FROM c_blkgrp_dist
GROUP BY region, metro
ORDER BY metro, region;
```

| region | metro | wgt_dist |
|---|---|---|
| 1 | M | 1.2 |
| 2 | M | 1.7 |
| 3 | M | 2.3 |
| 4 | M | 1.6 |
| 1 | R | 2.5 |
| 2 | R | 3 |
| 3 | R | 3.9 |
| 4 | R | 4.2 |

## Sum Population in Distance Bands

Count population within bands of distances; 0 to 1 mile, 1 to 2 miles, etc up to 6 miles and greater (shown below)

```
SELECT region, metro, SUM(pop2010) AS pop_6miles
FROM c_blkgrp_dist
WHERE dist > 6
GROUP BY region, metro
ORDER BY metro, region
```

| region | metro | pop_gt6miles |
|---|---|---|
| 1 | M | 389134 |
| 2 | M | 1298647 |
| 3 | M | 5131130 |
| 4 | M | 1529540 |
| 1 | R | 460374 |
| 2 | R | 2338454 |
| 3 | R | 5173601 |
| 4 | R | 1321819 |

## Percent Totals

Calculating % totals across rows is a pain. Export out of the database and calculate elsewhere.

```
SELECT t1.region, t1.metro, SUM(t1.pop2010) AS pop_6mile,
((SUM(cast(t1.pop2010 as real))) / t2.total) * 100 AS pct_6mile
FROM c_blkgrp_dist AS t1
JOIN (SELECT region, SUM(pop2010) AS total
  FROM c_blkgrp_dist
  WHERE metro='M'
  GROUP BY region) AS t2
ON t1.region=t2.region
WHERE t1.dist > 6 AND t1.metro='M'
GROUP BY t1.region, t1.metro
```

| region | metro | pop_6mile | pct_6mile |
|---|---|---|---|
| 1 | M | 389134 | 0.7792371528317198 |
| 2 | M | 1298647 | 2.5175307704947234 |
| 3 | M | 5131130 | 5.575822412153981 |
| 4 | M | 1529540 | 2.361452153660606 |

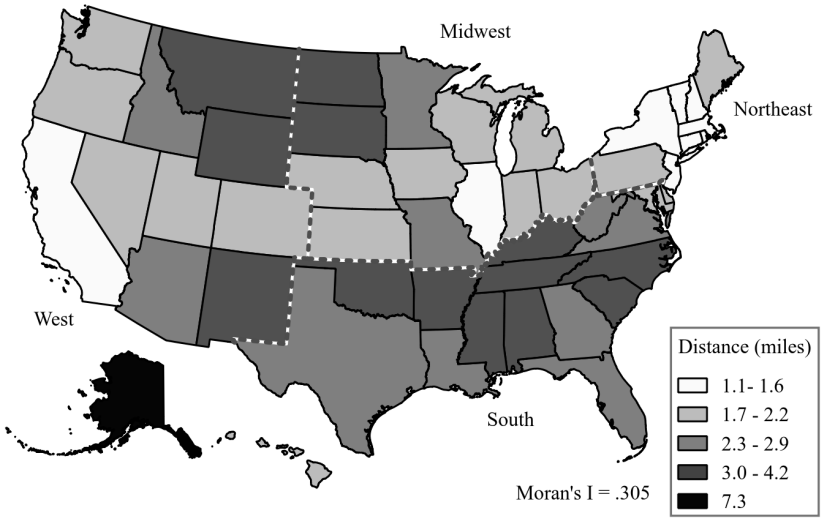
## Analyzing

- Export summaries out of the database, create percent totals in spreadsheet or use a scripting language
- Export the state attributes and distance measurements out to Excel to calculate correlations
- Export the state distance measurements out to a shapefile to run LISA calculations in OpenGEODA
- Use spatial views to create maps in QGIS

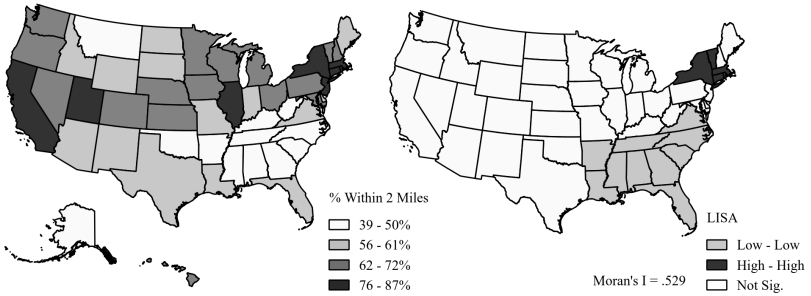
## Findings

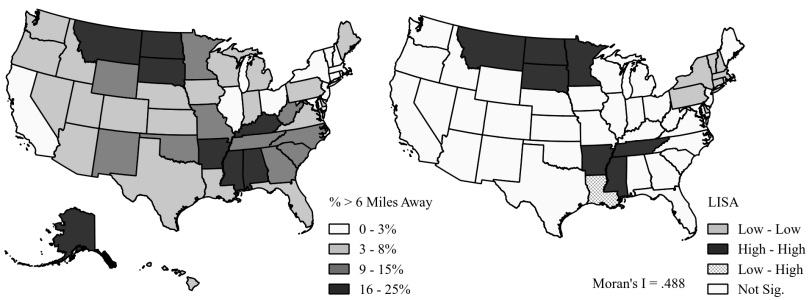
- The average American lives 2.1 miles from the nearest public library
- Approx 65% of Americans live within 2 miles of a library
- There is significant variation in library distance across states and regions
  - 68% of Metro pop vs 47% of non-metro pop lives within 2 miles of library
  - Pop in regions within 2 miles of library: NE 80%, MW 67%, S 52%, W 72%
- States that are more urban and densely populated tend to have shorter distances, but there are regional patterns that can't be explained by this alone

## Avg Distance to Nearest Library



## Population Within 2 Miles and More than 6 Miles



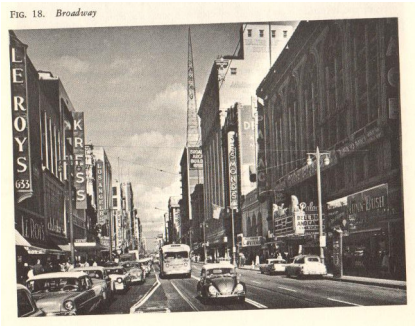



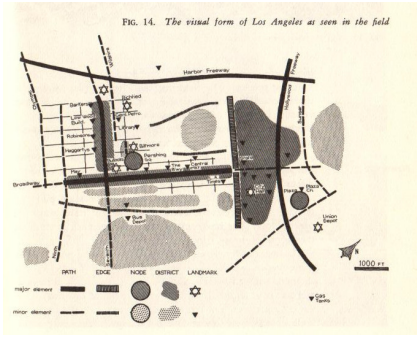

## Today's Topics



## Geometry and Storing Urban Features

Kevin Lynch *Image of the City* 1960. Physical elements of the city (it's form) and what it means to residents.



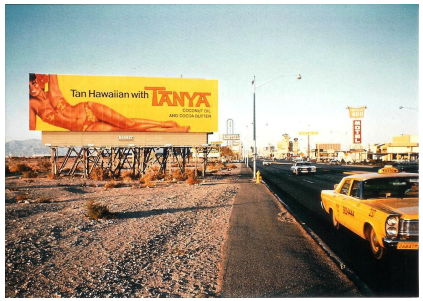

## City Elements

Lynch's elements:

- **Paths** Routes along which a person travels and observes the city, and from which the rest of the elements are arranged
- **Edges** Linear elements not considered as paths by the observer, they are boundaries or breaks in the landscape
- **Districts** Medium to large sections of the city that share some identifiable characteristics, and in which an observer moves into or out of
- **Nodes** Strategic points in a city that an observer can enter, they can be junctions, crossings, or simply concentrations
- **Landmarks** Points that serve as an identifiable physical object, but one that an observer typically does not enter into

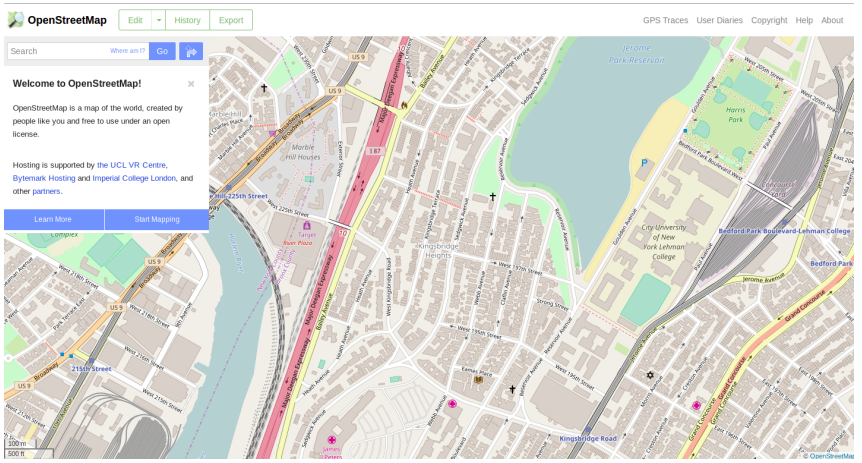
## City as Signs and Symbols

Venturi, Scott Brown & Izenour *Learning From Las Vegas* 1972 / 77. Styles and signs make connections between city elements.




## Open StreetMap

OSM as a collection of elements represented as geometries




## Spatial storage

Different approaches for storing geometry:

Heterogeneous: keep a mix of different geometries in one geometry column

- Declare column with generic type "Geometry": geometry(Geometry,SRID)

Homogeneous: only one type of geometry per geometry column

- Declare column as specific type (point, linestring, etc)
- Different tables for different geometries, or same table with separate geometry columns for each type

Inheritance: non-constrained parent table inherits fixed geometry of child tables

## Inheritance

Example in PostGIS in Action: a parent table with all roads, and child tables with rows for specific parts of the country. Inheritance gives you flexibility; query the parent to see everything or query one of the child tables for specifics.

1. Create empty parent table with necessary columns and data types
2. Create child tables with just a primary key and INHERITS the structure of the parent table
3. Alter the child tables to add check constraints to limit data based on attributes
4. Populate just the child tables with data

## OSM Data

- PostGIS in Action used some sleight of hand to import Open Street Map data into the sample database
- Used separate command-line tool osm2pgsql to map XML-based OSM data into a relational database structure
- `http://wiki.openstreetmap.org/wiki/Osm2pgsql`
- Used the special PostgreSQL hstore data type to store key-value pairs in a column called tags
- Mapped attributes from this column when needed
- tags ->'name' AS place_name

## OSM Data Structure

```
<node id="370353699" lat="48.8708079" lon="2.3033889"
user="Charlie Echo" uid="41390" visible="true" version="5"
changeset="4071729" timestamp="2010-03-08T13:19:26Z">
  <tag k="amenity" v="bicycle_rental"/>
  <tag k="capacity" v="38"/>
  <tag k="name" v="Champs-lyses Lincoln"/>
  <tag k="network" v="Vlib'"/>
  <tag k="ref" v="8041"/>
 </node>
 <node id="370560274" lat="48.8715167" lon="2.3000286"
 user="jihaire" uid="154300" visible="true" version="3"
 changeset="2529758" timestamp="2009-09-19T02:31:44Z">
  <tag k="amenity" v="bicycle_rental"/>
  <tag k="capacity" v="17"/>
  <tag k="FIXME" v="Station prsente mais non numerote"/>
  <tag k="name" v="39 rue de Bassano - 75016 Paris"/>
  <tag k="network" v="Vlib'"/>
 </node>
```

## Rules and Triggers

Refer to PostGIS in Action for details. Rules and triggers are objects that exist in the database, and have similar functionality.

### Rules

An instruction on how to rewrite a SQL statement. CREATE RULE somerule ON INSERT TO table DO INSTEAD something else. Rules are called once for each statement.

### Triggers

Prevents something from happening if certain conditions aren't met, does something instead of a requested INSERT, UPDATE, DELETE command, or does something else in addition to one of those commands. Triggers are called for each row.

## Is Your Geometry in Good Shape?

Use ST_Equals(a.geometry, b.geometry) to test whether two geometries are equivalent, and ST_IsValid(geometry) to verify whether geometry (polygons) is properly formed (no overlapping boundaries, rings closed, etc).

```
SELECT zcta, ST_IsValid(geom), ST_IsValidReason(geom)
FROM nyc.zctas
WHERE NOT ST_IsValid(geom);
```

Use ST_MakeValid(geometry) to try to repair geometries, but be careful - making a backup is a good idea.

## Next Class

The following are due at the beginning of our next class:

Assignment #10

Posted on the course website
Will be returned to you via email by Tue Apr 23rd at the latest

Readings for Class #11

Listed in the syllabus, in the *Practical SQL* book