Spatial Database Management
GEP 664 / GEP 380
Class #8: Spatial relationships and analysis

Frank Donnelly

Dept of EEGS, Lehman College CUNY

Spring 2019

Today's Topics

Indexes

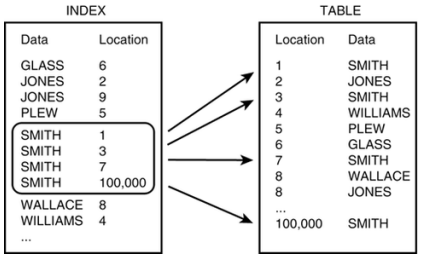


Image source:
https://en.wikipedia.org/wiki/Database_index

- Speeds up access based on a specific column
- Keys are indexed by default
- Values can be unique or not (unique is best)
- Bad for small tables or small set of values

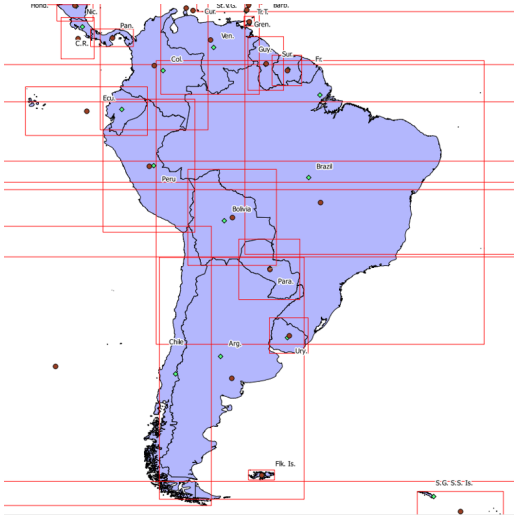
```
CREATE INDEX indexname
ON tablename (column);
```

Spatial Index

- Indexes are db objects. They are typically hierarchial trees that speed up searching and selection for specific columns.
- In a spatial index, bounding boxes of geometry are stored in the index and are checked first when spatial functions are performed.
- PostGIS uses the gist index most commonly (generalized search tree)
- Spatial index is created by default when shapefiles are loaded using PostGIS

Bounding Boxes

The smallest rectangular box that can be drawn around an individual feature that completely encloses it.



Creating Spatial Index Manually

```
CREATE INDEX idx_census_tracts_geom
ON nyc.census_tracts
USING gist (geometry);
```

Relating Geometries

There are several functions that will return different results based on how different geometries relate. Below: the geometries on the left TOUCH while the ones on the right OVERLAP. Both of them INTERSECT.

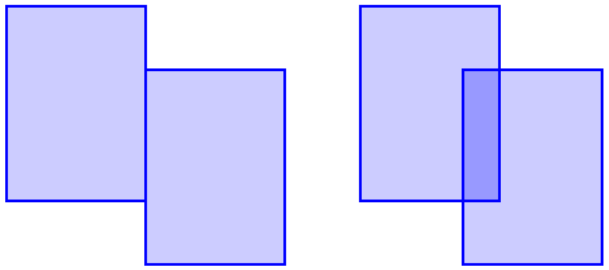


Image source: http://daniel-zuma.com/img/georails/county_boundaries.png

Relationship Criteria

When relating geometries they must share the same SRS and be valid geometries. 2D geometry functions rely on these concepts - for every geometry they check these relationships:

- **Interior:** space inside the geometry and not on the boundary
- **Exterior:** space outside the geometry and not on the boundary
- **Boundary:** space that's not interior or exterior

Intersects

ST_Intersects (returns true/false)

Intersects - when geometry A intersects geometry B:

- A has interior or boundary points in common with B
- The broadest, and likely the most common, operation

If A intersects B then B intersects A.

Sample Statement

Return all census tracts that intersect subway stations (use DISTINCT to remove duplicates; tracts that have multiple stations).

```
SELECT DISTINCT c.tractid, c.name
FROM nyc.census_tracts c, nyc.subway_stations s
WHERE ST_Intersects (c.geometry, s.geometry);
```

Return all census tracts that intersect subway stations and count the number of stations.

```
SELECT c.tractid, c.name, COUNT(s.stop_id) AS stations
FROM nyc.census_tracts c, nyc.subway_stations s
WHERE ST_Intersects (c.geometry, s.geometry)
GROUP BY c.tractid, c.name
ORDER BY stations DESC, c.tractid;
```

Contains and Within

ST_Contains and ST_Within

Contains - when geometry A contains geometry B:

- No points of B can lie in the exterior of A
- At least one point of B must lie in the interior of A
- Because of the previous point, if geometry B coincides perfectly with the boundaries of geometry A, A would NOT contain B

Within - is the inverse of contain. If geometry A is within geometry B, then geometry B contains geometry A.

Covers and Covered By

ST_Covers and ST_CoveredBy

Has similar criteria as Contains and Within, except:

- Covers also includes geometries where geometry B coincides perfectly with the boundaries of geometry A.

CoveredBy - is the inverse of covers. If geometry A is covered by geometry B, then geometry B covers geometry A.

Overlaps

ST_Overlaps

Overlap - geometry A overlaps with B when:

- A and B share some interior space
- The geometry of one is not completely contained within the other
- The geometries are of the same type

If A overlaps B then B overlaps A.

Touch

ST_Touches

Touch - geometry A touches geometry B when:

- A and B have at least one boundary point in common
- None of the common points lie in the interior of A or B

If A touches B then B touches A.

Cross

ST_Crosses

Cross - geometry A crosses geometry B when:

- A and B have some interior points in common but not all
- This implies that geometries that cross cannot touch or contain each other
- Only applicable for lines (crossing other lines or polygons) or multipoints (crossing lines or polygons)

If A crosses B then B crosses A.

Disjoint

ST_Disjoint

Disjoint- geometry A is disjoint with B when:

- A and B have no shared interiors or boundaries

ST_Disjoint can be an inefficient operation because it cannot use the spatial index. An alternative is to run an ST_Intersect statment with a LEFT JOIN to return NULL values (i.e. where geometry of B doesn't intersect with A).

Equality

Spatial equality

Uses ST_Equals. Means that two geometries occupy the same space.
WHERE ST_Equals(geom1,geom2)

Geometric equality

Uses ST_OrderingEquals. This is stricter: geometries must share the same space, the same type, and the same ordering of points (i.e. line that starts at point A and ends at point B is not the same as line starting at B and ending at A).

Bounding Box Equality

Uses the equals sign =. The bounding boxes of the two geometries share the same space.
WHERE geom1=geom2

Functions and Geometry Selections

Combine different functions to replicate additional geometry selection operations that you would find in GIS software.

Select all geometries that have their geographic center within other geometries (typically used for polygons):

```
SELECT c.tractid, c.tractname
FROM nyc.census_tracts c, nyc.zctas z
WHERE ST_Within(ST_Centroid(c.geometry),z.geom)
AND z.zcta='10010';
```

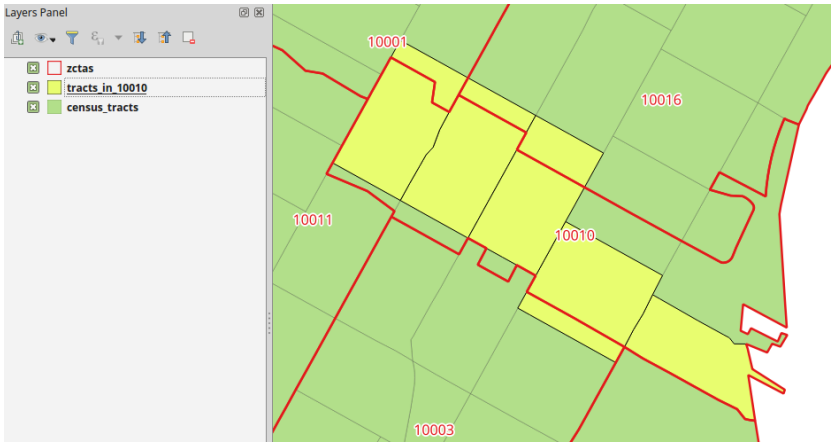
	tractid character varying(11)	tractnum character varying(6)	geometry geometry(MultiPolygon,2263)
1	36061005600	005600	0106000020D7080000010
2	36061005800	005800	0106000020D7080000010
3	36061006000	006000	0106000020D7080000010
4	36061006400	006400	0106000020D7080000010
5	36061006800	006800	0106000020D7080000010

Visualizing Results

- To see the actual results in GIS, the geometry of the features you want to return must be included in the SELECT clause.
- If you write the statement within the PostgreSQL interface, you would save it as a View that you can add to GIS.
- If you write the statement in the GIS DB interface, you can simply create a temporary object when executing the statement.

```
SELECT c.tractid, c.tractname, c.geometry
FROM nyc.census_tracts c, nyc.zctas z
WHERE ST_Within(ST_Centroid(c.geometry),z.geom)
AND z.zcta='10010';
```

Results Symbolized in QGIS



Selecting Geometry Within the Same Layer

For many of the geometry selection functions it's assumed that you're selecting geometry between layers. But it's also possible to select geometry within a layer using aliases.

Example of creating a neighbor list of census tracts:

```
SELECT t1.tractid AS tract, t2.tractid AS neighbor
FROM nyc.tracts AS t1, nyc.tracts AS t2
WHERE ST_Touches (t1.geom, t2.geom)
ORDER BY tract, neighbor;
```

Today's Topics

Relational versus Spatial Join

To join tables in a relational db, they must share a common attribute. To join tables in a spatial db, you can use the geometry to associate them. Assign subway stations the id of the census tract in which they are located.

1st approach - using WHERE

```
SELECT s.stop_id, s.stop_name, s.trains, c.tractid, c.tractnum
FROM nyc.subway_stations s, nyc.census_tracts c
WHERE ST_Within (s.geometry, c.geometry);
```

2nd approach - using JOIN

```
SELECT s.stop_id, s.stop_name, s.trains, c.tractid, c.tractnum
FROM nyc.subway_stations s
JOIN nyc.census_tracts c
ON ST_Within (s.geometry, c.geometry);
```

Saving Join Results

Create a View, or create a new table and load data, or add a column and update the existing table.

```
ALTER TABLE nyc.subway_stations
ADD COLUMN ctract varchar(11);
```

```
UPDATE nyc.subway_stations AS s
SET ctract = c.tractid
FROM nyc.census_tracts AS c
WHERE ST_Within (s.geometry, c.geometry);
```

Geoprocessing - Dis-aggregating

Use ST_Dump to break multi-part features into single-part features:

```
SELECT tractid, tractnum, bcode, ST_Dump(geometry) AS
    newgeom
FROM nyc.census_tracts;
```

	tractid character varying(11)	tractnum character varying(6)	bcode character varying(5)	newgeom geometry_dump
1	36005000100	000100	36005	({1},0103000020D708000001000
2	36005000200	000200	36005	({1},0103000020D708000001000
3	36005000400	000400	36005	({1},0103000020D708000001000
4	36005000400	000400	36005	({2},0103000020D708000001000
5	36005000400	000400	36005	({3},0103000020D708000001000
6	36005001600	001600	36005	({1},0103000020D708000001000

Geoprocessing - Aggregating

Use ST_Union to group features together. Example of grouping features that share the same attribute:

```
SELECT nta, bcode, ST_Union(geometry) AS newgeom
FROM nyc.census_tracts
GROUP BY nta, bcode;
```

Census Tracts (Before)

NTAs (After)

Note - some GIS packages refer to this operation as Dissolve.

Making the New Geometry Permanent

Create a new table and insert the data to make it permanent.

```
CREATE TABLE nyc.nbhoods (
nta varchar(4) PRIMARY KEY,
bcode varchar(5),
geom geometry(multipolygon,2263));
```

Here we have to use ST_Multi, to insure that any single polygons are stored as multipolygons.

```
INSERT INTO nyc.nbhoods (nta,bcode,geom)
SELECT nta, bcode, ST_Multi(ST_Union(geometry))
FROM nyc.census_tracts
GROUP BY nta, bcode;
```

Otherwise, you could just create a View.

Geoprocessing - Intersection

ST_Interesction returns actual geometry that Intersects (different from ST_Intersects, which returns true / false)

Calculate % of census tracts' area that intersects ZCTA 10010:

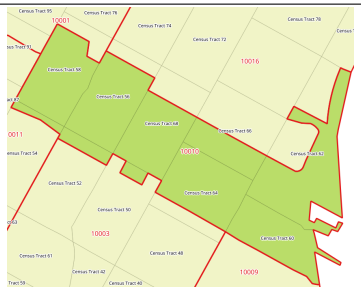
```
SELECT c.tractnum,
ROUND (ST_Area(ST_Intersection(c.geometry,z.geom)) /
   ST_Area(c.geometry)*100) AS pct_in
FROM nyc.census_tracts c, nyc.zctas z
WHERE ST_Intersects(c.geometry, z.geom)
AND z.zcta='10010'
ORDER BY pct_in DESC;
```

	tractnum character varying(6)	pct_in double precision
1	006000	100
2	005600	86
3	005800	82
4	006400	81
5	006800	72

Intersection Spatial View

Dump to break tract geometry apart, use overlaps / within instead of intersects to exclude areas that touch:

```
CREATE VIEW nyc.tract_10010_ovlap AS
SELECT c.tractnum, (ST_Dump(ST_Intersection(c.geometry,z.
    geom))).geom AS newgeom
FROM nyc.census_tracts c, nyc.zctas z
WHERE z.zcta='10010' AND
(ST_Overlaps(c.geometry, z.geom) OR ST_Within(c.geometry, z.
    geom))
```



Other Geoprocessing Features

- Difference (ST_Difference returns portion of geom A not shared with geom B)
- Symmetric difference (ST_SymDifference returns geom of A and B that's not shared)
- Split (ST_Split)
- Tessellate (creates grid)

Today's Topics

Due Next Class

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

Assignment #8

Posted on the course website

Midterm Quiz

Takes place at the beginning of class

Readings for Class #9

Listed in the syllabus, in the *PostGIS In Action* book

PostGIS in Action Readings

READ Chapter 10