Chapter 5/7

Spatial DBMS

Spatial Database

- Spatial databases provide structures for storage and analysis of spatial data
- Spatial data is comprised of objects in multi-dimensional space
- Storing spatial data in a standard database would require excessive amounts of space
- Queries to retrieve and analyze spatial data from a standard database would be long and cumbersome leaving a lot of room for error
- Spatial databases provide much more efficient storage, retrieval, and analysis of spatial data

Types of data stored on Spatial Database:

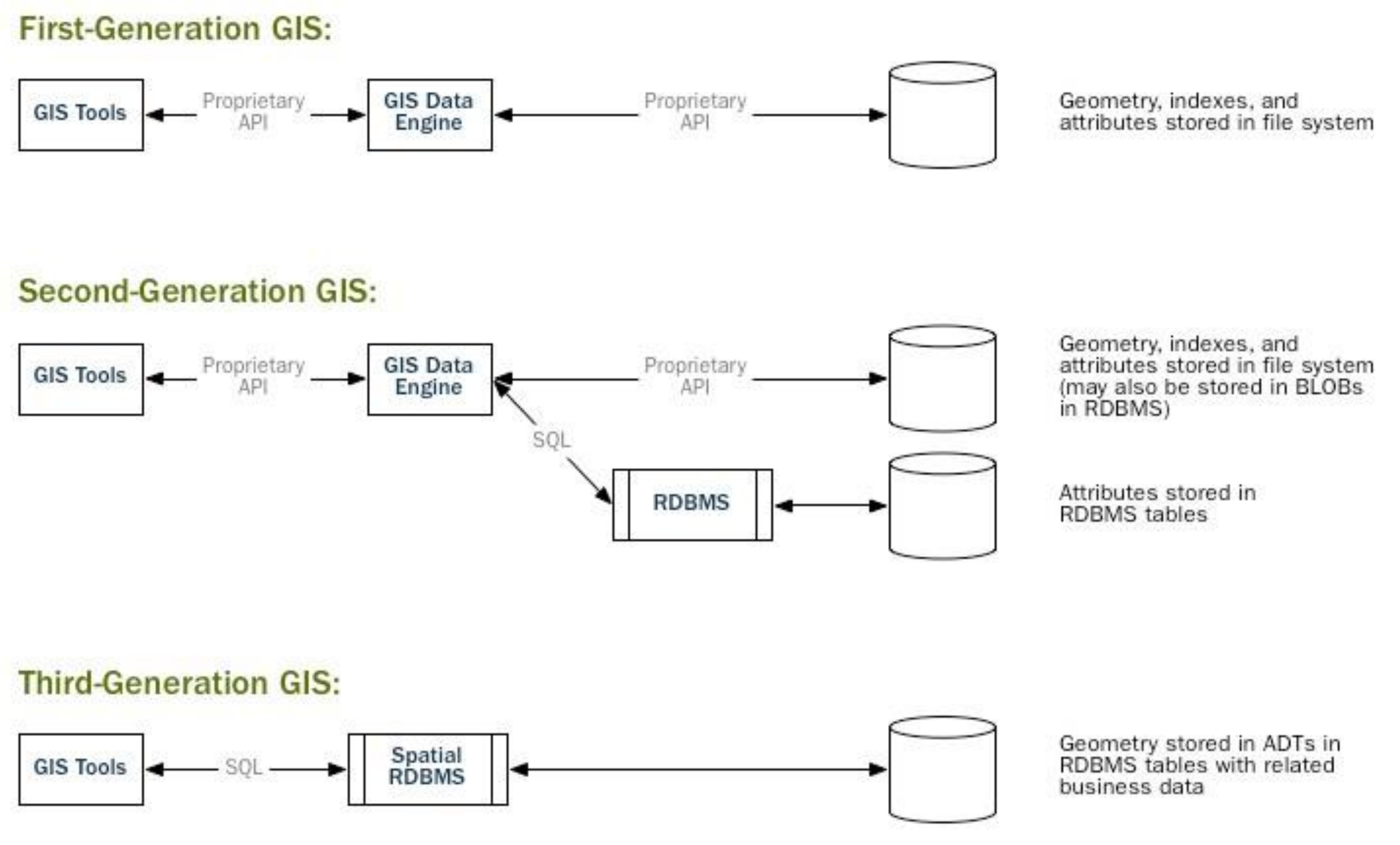
Two-dimensional Data	Three-dimensional Data
Geographical	Weather
Cartesian coordinates (2D)	Cartesian coordinates (3D)
Network	Topological
Direction	Satellite Images

Spatial Data

- Customer Location
- Store Locations
- Transportation Tracking
- Statistical/Demographic
- Epidemiology
- Crime Patterns

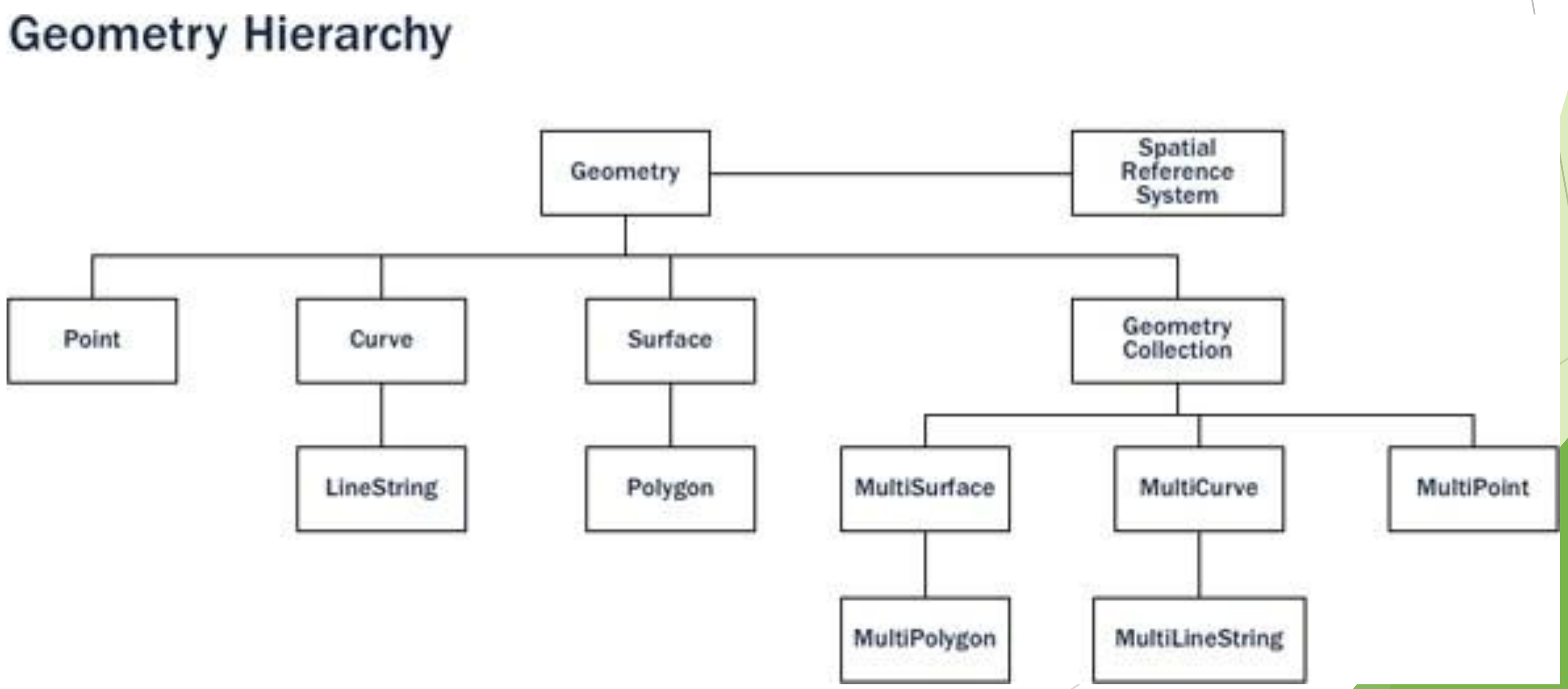
- Weather Information
- Land Holding
- Natural Resources
- City Planning
- Environmental Planning
- Hazard Detection
- Cell phone tracking

Spatial Database



Spatial Data Types

- An ordinary database has strings, numbers, and dates.
- A spatial database adds additional (spatial) types for representing geographic features.
- These spatial data types abstract and encapsulate spatial structures such as boundary and dimension.
- In many respects, spatial data types can be understood simply as shapes.

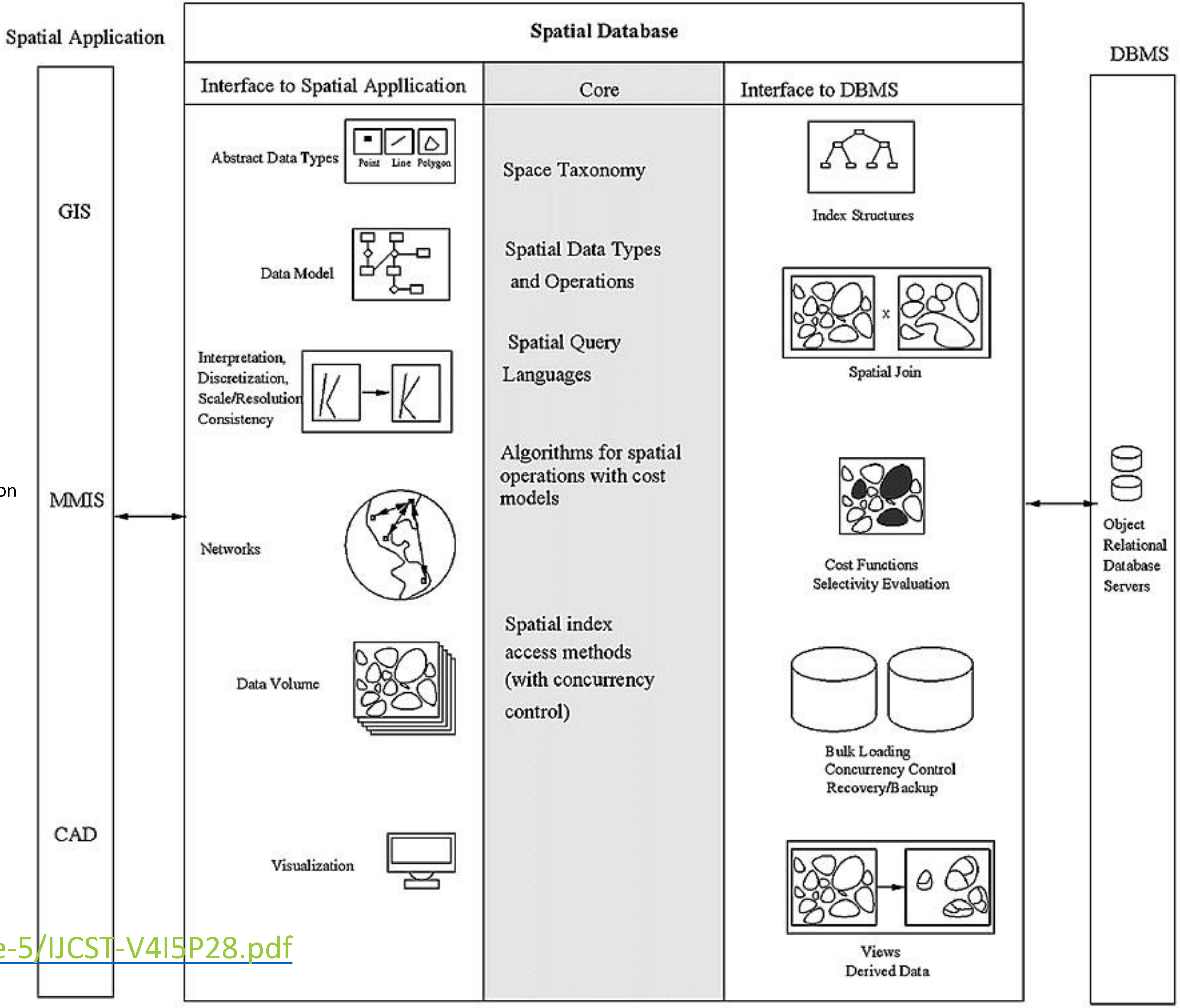


Spatial Database Management System

- Spatial Database Management System (SDBMS) provides the capabilities of a traditional database management system (DBMS) while allowing special storage and handling of spatial data.
- SDBMS Works with an underlying DBMS
- It offers spatial data types/data models/ query language
  - Support spatial properties/operations
- It supports spatial data types in its implementation
  - Support spatial indexing, algorithms for spatial selection and join

Architecture

Multimedia Information System



http://www.ijcstjournal.org/volume-4/issue-5/IJCST-V4I5P28.pdf

Advantages of Spatial Databases

- Able to treat spatial data like anything else in the database.
- Offset complicated tasks to the DB server
  - Organization and indexing done for you
  - Do not have to re-implement operators
  - Do not have to re-implement functions
- Significantly lowers the development time of client applications
  - Spatial querying using SQL
- Use simple SQL expressions to determine spatial relationships
  - Distance, containment
- Use simple SQL expressions to perform spatial operations
  - Area, length, intersection, union, buffer

Disadvantages of Spatial Databases

- Cost to implement can be high
- Some inflexibility
- Incompatibilities with some GIS software
- Slower than local, specialized data structures

Spatial Database Offerings

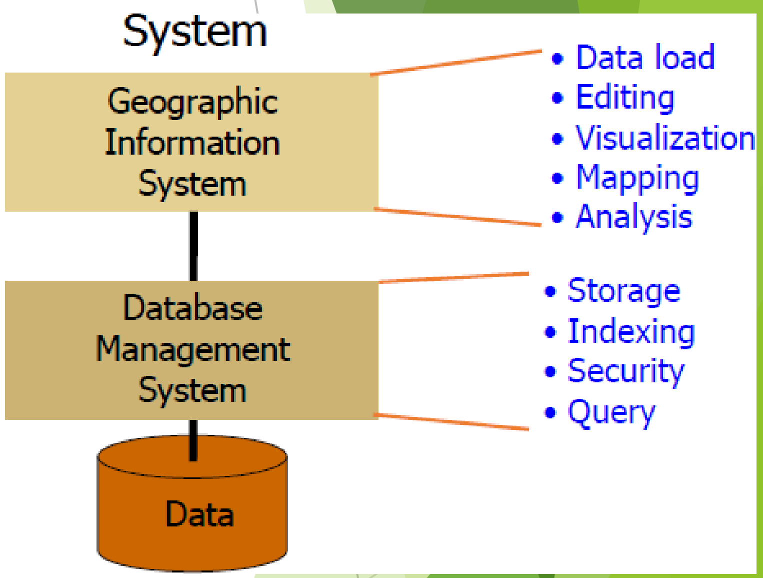
- ESRI ArcSDE (on top of several different DBs)
- Oracle Spatial
- IBM DB2 Spatial Extender
- Informix Spatial DataBlade
- MS SQL Server (with ESRI SDE)
- Geomedia on MS Access
- MySQL Spatial
- PostGIS / PostgreSQL

SDBMS different from GIS

- GIS is a software to visualize and analyze spatial data using spatial analysis functions such as
  - Search: Thematic search, search by region, (re-)classification
  - Location analysis: Buffer, corridor, overlay
  - Terrain analysis: Slope/aspect, catchment, drainage network
  - Flow analysis: Connectivity, shortest path
  - Distribution: Change detection, proximity, nearest neighbor
  - Spatial analysis/Statistics: Pattern, centrality, autocorrelation, indices of similarity, topology: hole description
  - Measurements: Distance, perimeter, shape, adjacency, direction
- GIS uses SDBMS
  - to store, search, query, share large spatial data sets

SDBMS different from GIS

- SDBMS focusses on
  - Efficient storage, querying, sharing of large spatial datasets
  - Provides simpler set-based query operations
  - Example operations: search by region, overlay, nearest neighbor, distance, adjacency, perimeter etc.
  - Uses spatial indices and query optimization to speedup queries over large spatial datasets.
- SDBMS may be used by applications other than GIS
  - Urban planning, route optimization, fire or pollution monitoring, utility networks.



SDBMS

- Consider a spatial dataset with:
  - County boundary (dashed white line)
  - Census block -name, area, population, boundary (dark
  - Water bodies (dark polygons)
  - Satellite Imagery (gray scale pixels)

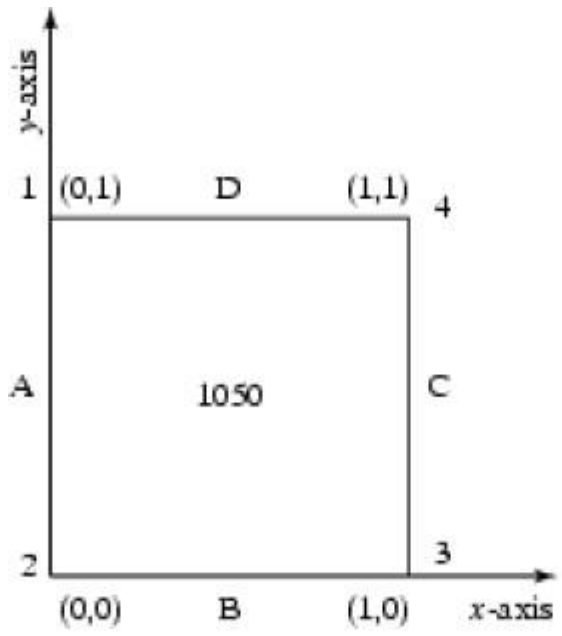
- Storage in a SDBMS table:

```
create table census_blocks (
name          string,
area          float,
population    number,
boundary      polygon );
```

SDBMS

Modeling Spatial Data in Traditional DBMS

- A row in the table census_blocks
- Is Polyline datatype supported in DBMS?



Census_blocks

Name	Area	Population	Boundary
1050	1	1839	Polyline((0,0),(0,1),(1,1),(1,0))

SDBMS

Modeling Spatial Data in Traditional DBMS

Census_blocks

Name	Area	Population	boundary-ID
340	1	1839	1050

Polygon

boundary-ID	edge-name
1050	A
1050	B
1050	C
1050	D

Edge

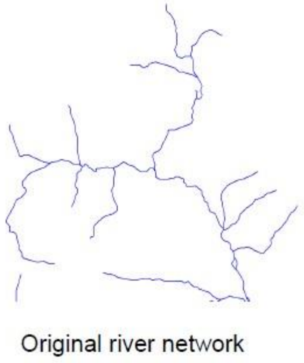
edge-name	endpoint
A	1
A	2
B	2
B	3
C	3
C	4
D	4
D	1

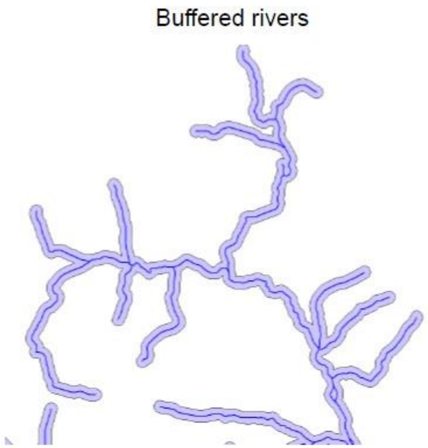
Point

endpoint	*x*-coor	*y*-coor
1	0	1
2	0	0
3	1	0
4	1	1

Spatial Query Language Chapter 7

- An interface and retrieval language between a user and a database system that includes spatial operations.
- A formal language that allows formulating spatial queries by providing topological, directional and metric operators for specifying selection criteria.





Basic operations on all data types

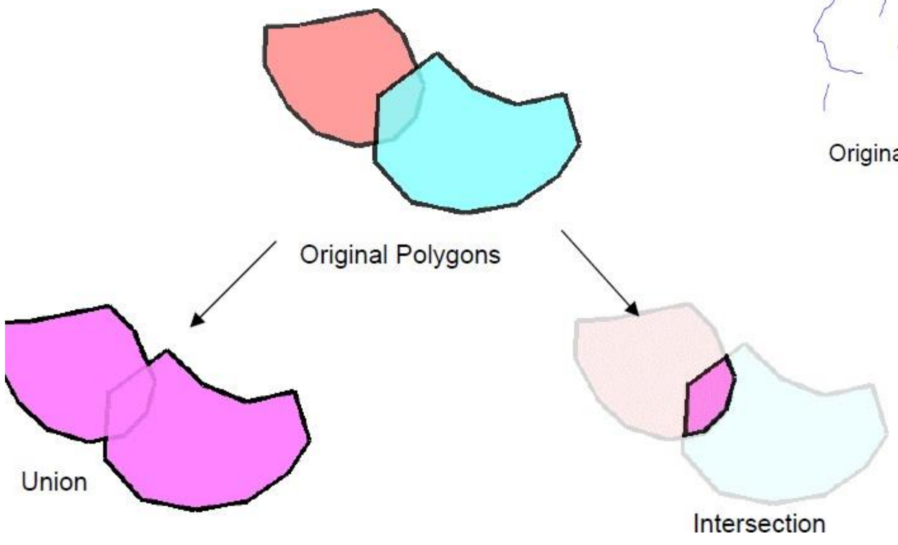
- Ex: IsEmpty, Envelope, Boundary

Topological/set operators

- Ex: Disjoint, Touch, Contains

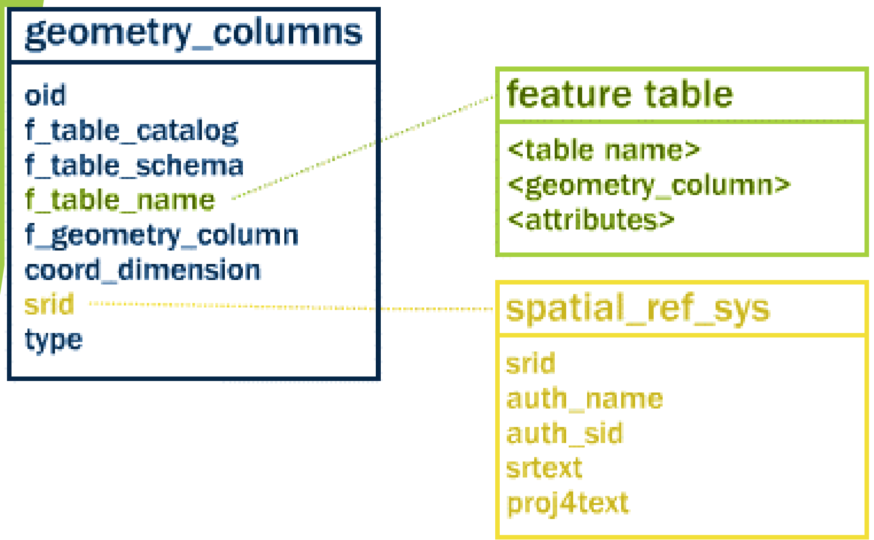
Spatial analysis

- Ex: Distance, Intersection



Geometries-Example

CREATE a table (geometries) then INSERTs five geometries: a point, a line, a polygon, a polygon with a hole, and a collection. Finally, the inserted rows are selected and displayed in the Output pane.




```
CREATE TABLE geometries (name varchar, geom geometry);

INSERT INTO geometries VALUES
  ('Point', 'POINT(0 0)'),
  ('Linestring', 'LINESTRING(0 0, 1 1, 2 1, 2 2)'),
  ('Polygon', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))'),
  ('PolygonWithHole', 'POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),(1 1, 1 2, 2 2, 2 1, 1 1))'),
  ('Collection', 'GEOMETRYCOLLECTION(POINT(2 0),POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))');

SELECT name, ST_AsText(geom) FROM geometries;
```

```
SELECT * FROM geometry_columns;
```

f_table_catal character var	f_table_schei character var	f_table_name character varying(256	f_geometry_ character var	coord_dimen: integer	srid integer	type character varying(
	public	nyc_census_blocks	the_geom	2	26918	MULTIPOLYGON
	public	nyc_neighborhoods	the_geom	2	26918	MULTIPOLYGON
	public	nyc_streets	the_geom	2	26918	MULTILINESTRING
	public	nyc_subway_stations	the_geom	2	26918	POINT
	public	geometries	geom	2	-1	POINT

Geometries-Example

Our example table contains a mixture of different geometry types. We can collect general information about each object using functions that read the geometry metadata.

ST_GeometryType(geometry) returns the type of the geometry
ST_NDims(geometry) returns the number of dimensions of the geometry
ST_SRID(geometry) returns the spatial reference identifier number of the geometry

- SELECT name, ST_GeometryType(geom), ST_NDims(geom), ST_SRID(geom)

  FROM geometries;

	name character varying	st_geometrytype text	st_ndims smallint	st_srid integer
1	Point	ST_Point	2	0
2	Linestring	ST_LineString	2	0
3	Polygon	ST_Polygon	2	0
4	PolygonWithHole	ST_Polygon	2	0
5	Collection	ST_GeometryCollection	2	0

- SELECT ST_AsText(geom) FROM geometries WHERE name = 'Point';
- SELECT ST_X(geom), ST_Y(geom) FROM geometries WHERE name = 'Point';

st_astext text
POINT(0 0)

st_x double precision	st_y double precision
0	0

Geometries-Linestrings

- Some of the specific spatial functions for working with linestrings

ST_Length(geometry) returns the length of the linestring **ST_StartPoint(geometry)** returns the first coordinate as a point **ST_EndPoint(geometry)** returns the last coordinate as a point **ST_NPoints(geometry)** returns the number of coordinates in the linestring

- SELECT ST_AsText(geom) FROM geometries WHERE name = 'Linestring';
- SELECT ST_Length(geom) FROM geometries WHERE name = 'Linestring';

Simple non-closed linestring

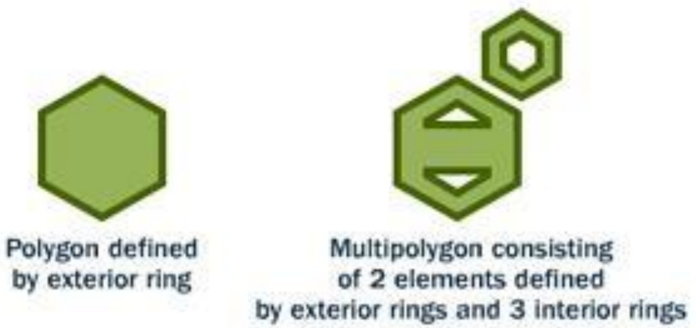
Simple multilinestring defined by 4 endpoints of 2 elements

Geometries-Polygons

- Some of the specific spatial functions for working with Polygons

ST_Area(geometry) returns the area of the polygons
ST_NRings(geometry) returns the number of rings (usually 1, more of there are holes)
ST_ExteriorRing(geometry) returns the outer ring as a linestring
ST_InteriorRingN(geometry,n) returns a specified interior ring as a linestring
ST_Perimeter(geometry) returns the length of all the rings

- SELECT ST_AsText(geom) FROM geometries WHERE name LIKE 'Polygon%';
- SELECT name, ST_Area(geom) FROM geometries WHERE name LIKE 'Polygon%';



Geometry Input and Output

- Well-known text (WKT)
  - ST_GeomFromText(text, srid) returns geometry
  - ST_AsText(geometry) returns text
  - ST_AsEWKT(geometry) returns text
- Well-known binary (WKB)
  - ST_GeomFromWKB(bytea) returns geometry
  - ST_AsBinary(geometry) returns bytea
  - ST_AsEWKB(geometry) returns bytea
- Geographic Mark-up Language (GML)
  - ST_GeomFromGML(text) returns geometry
  - ST_AsGML(geometry) returns text
- Keyhole Mark-up Language (KML)
  - ST_GeomFromKML(text) returns geometry
  - ST_AsKML(geometry) returns text
- GeoJSON
  - ST_AsGeoJSON(geometry) returns text
- Scalable Vector Graphics (SVG)
  - ST_AsSVG(geometry) returns text

https://postgis.net/workshops/postgis-intro/glossary.html#term-wkt

- "Well-known text" Can refer either to the text representation of geometries, with strings starting "POINT", "LINESTRING", "POLYGON", etc.
- "Well-known binary" Refers to the binary representation of geometries described in the Simple Features for SQL specification (SFSQL).
- GML Geography Markup Language. GML is the OGC standard XML format for representing spatial feature information.
- KML "Keyhole Markup Language", the spatial XML format used by Google Earth. Google Earth was originally written by a company named "Keyhole", hence the (now obscure) reference in the name.
- GeoJSON "Javascript Object Notation", a text format that is very fast to parse in Javascript virtual machines. In spatial, the extended specification for GeoJSON is commonly used.
- SVG"Scalable vector graphics" is a family of specifications of an XML-based file format for describing two-dimensional vector graphics, both static and dynamic (i.e., interactive or animated).

Examples of Spatial Query

1. What is the area of the 'West Village' neighborhood?
   - SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name = 'West Village';
2. What is the area of Manhattan in acres?
   - SELECT Sum(ST_Area(geom)) / 4047 FROM nyc_neighborhoods

     WHERE boroname = 'Manhattan';
3. What is the total length of streets (in kilometers) in New York City?
   - SELECT Sum(ST_Length(geom)) / 1000 FROM nyc_streets;
4. What is the JSON representation of the boundary of the 'West Village'?
   - SELECT ST_AsGeoJSON(geom) FROM nyc_neighborhoods WHERE name = 'West Village';
5. What is the length of streets in New York City, summarized by type
   - SELECT type, Sum(ST_Length(geom)) AS length FROM nyc_streets GROUP BY type ORDER BY length DESC;

Spatial Query - Join

- Spatial join example

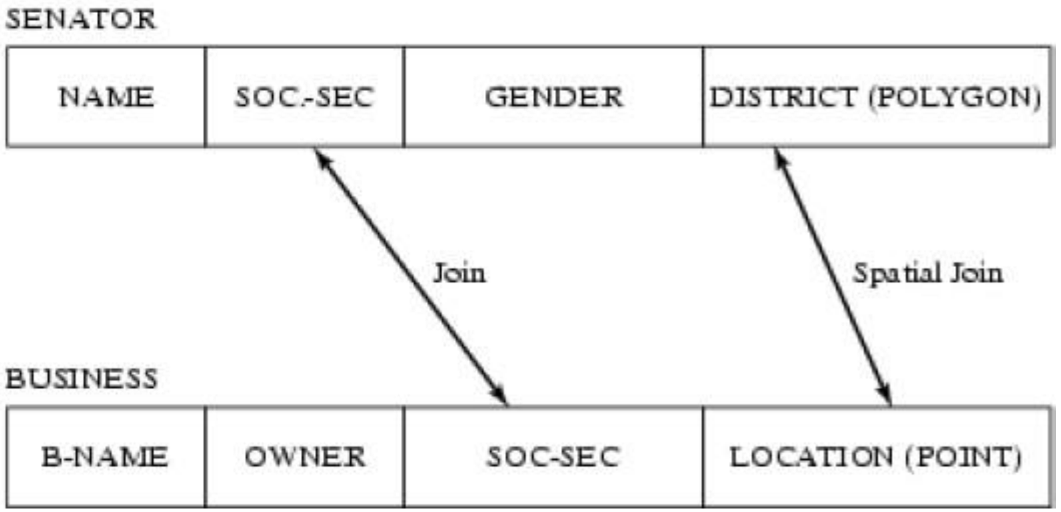
  SELECT S.name FROM Senator S, Business B WHERE

  S.district.Area() > 300 AND Within(B.location, S.district)

- Non-Spatial Join example

  SELECT S.name FROM Senator S, Business B

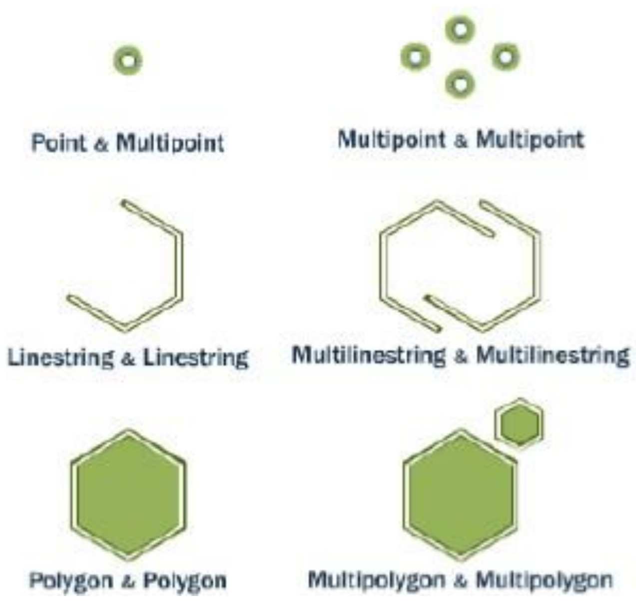
  WHERE S.soc-sec = B.soc-sec AND S.gender = ‘Female’



Spatial Relationships

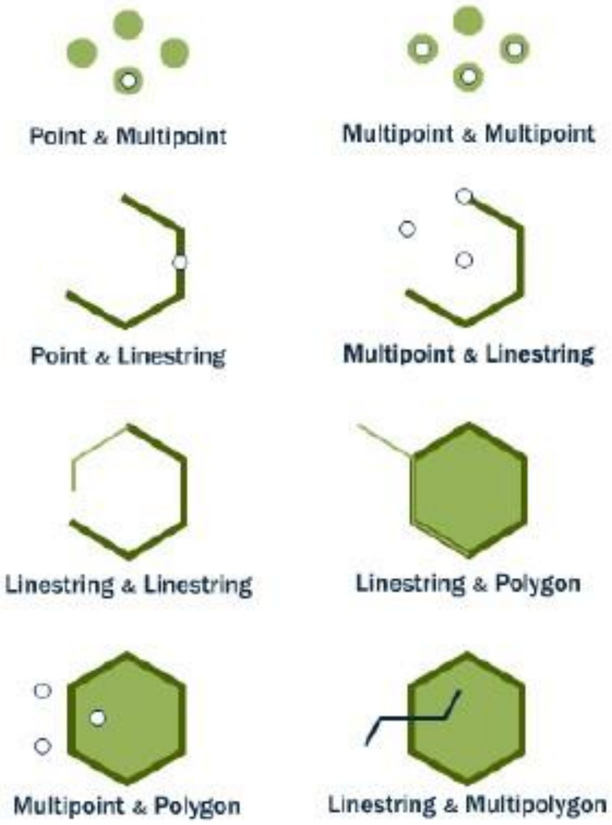
- **ST_Equals(geometry A, geometry B)** tests the spatial equality of two geometries. ST_Equals returns TRUE if two geometries of the same type have identical x,y coordinate values, i.e. if the second shape is equal (identical) to the first shape.
  - SELECT name, geom, ST_AsText(geom) FROM nyc_subway_stations WHERE name = 'Broad St';
  - SELECT name FROM nyc_subway_stations

    WHERE ST_Equals(geom, '0101000020266900000EEBD4CF27CF2141BC17D69516315141');



Spatial Relationships

- **ST_Intersects(geometry A, geometry B)** returns t (TRUE) if the two shapes have any space in common, i.e., if their boundaries or interiors intersect.

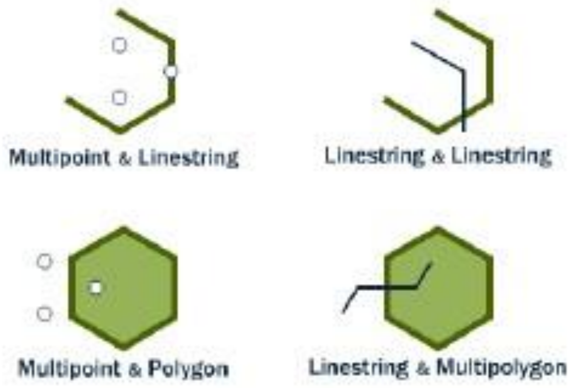


Spatial Relationships

- **ST_Disjoint(geometry A , geometry B).** If two geometries are disjoint, they do not intersect, and vice-versa. In fact, it is often more efficient to test “not intersects” than to test “disjoint” because the intersects tests can be spatially indexed, while the disjoint test cannot.

Spatial Relationships

- **ST_Crosses(geometry A, geometry B)** returns t (TRUE) if the intersection results in a geometry whose dimension is one less than the maximum dimension of the two source geometries and the intersection set is interior to both source geometries.



- **ST_Overlaps(geometry A, geometry B)** compares two geometries of the same dimension and returns TRUE if their intersection set results in a geometry different from both but of the same dimension

Spatial Relationships

- **ST_Touches** tests whether two geometries touch at their boundaries, but do not intersect in their interiors

- **ST_Within(geometry A , geometry B)** returns TRUE if the first geometry is completely within the second geometry. ST_Within tests for the exact opposite result of ST_Contains.
- **ST_Contains(geometry A, geometry B)** returns TRUE if the second geometry is completely contained by the first geometry.

Spatial Relationships

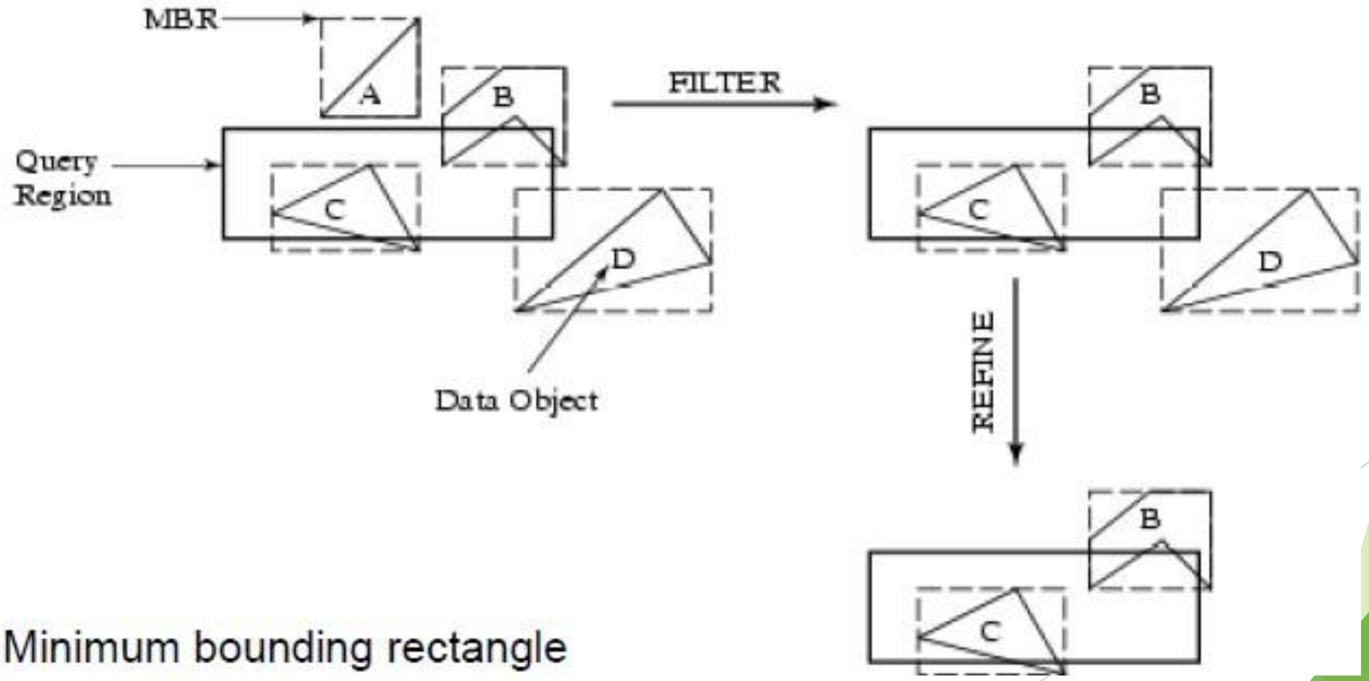
- An extremely common GIS question is “find all the stuff within distance X of this other stuff”.
- **ST_Distance(geometry A, geometry B)** calculates the shortest distance between two geometries and returns it as a float. This is useful for actually reporting back the distance between objects.
- For testing whether two objects are within a distance of one another, the **ST_DWithin** function provides an index-accelerated true/false test. This is useful for questions like “how many trees are within a 500 meter buffer of the road?”.

- Using our Broad Street subway station again, we can find the streets nearby (within 10 meters of) the subway stop:
  - SELECT name FROM nyc_streets

    WHERE ST_DWithin(geom, ST_GeomFromText('POINT(583571 4506714)',26918), 10);

Query Processing

- Efficient algorithms to answer spatial queries
- Common Strategy - filter and refine
  - Filter Step: Query Region overlaps with MBRs of B,C and D
  - Refine Step: Query Region overlaps with B and C



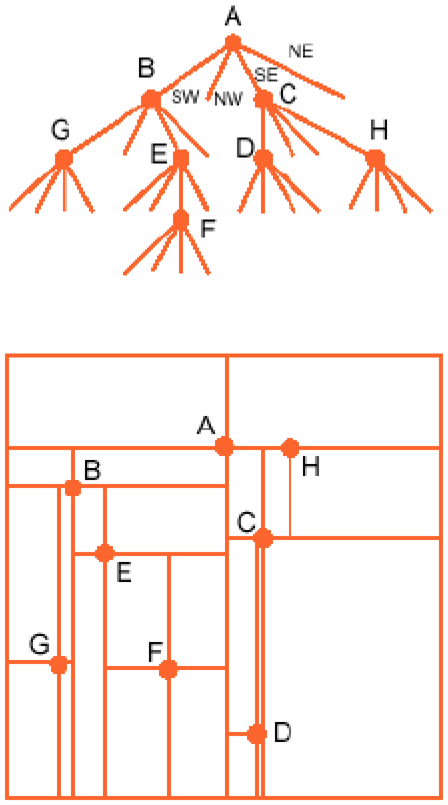
MBR: Minimum bounding rectangle

Spatial Index

- Used to locate rows quickly
  - Like a book index, it is a special representation of the content that adds order and makes finding items faster
- RDBMS use simple 1-d indexing
- Spatial DBMS needs 2-d, hierarchical indexing
  - Grid
  - Quadtree
  - R-tree
- Multi-level queries often used for performance (MBR)

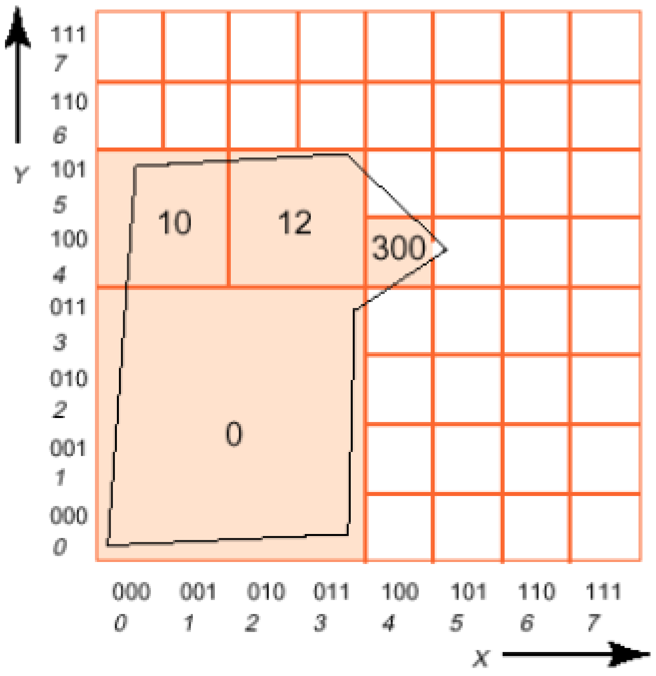
Spatial Index

- Point and Region Quadtree Indexing



Point Quadtree

Based on recursive division of space.

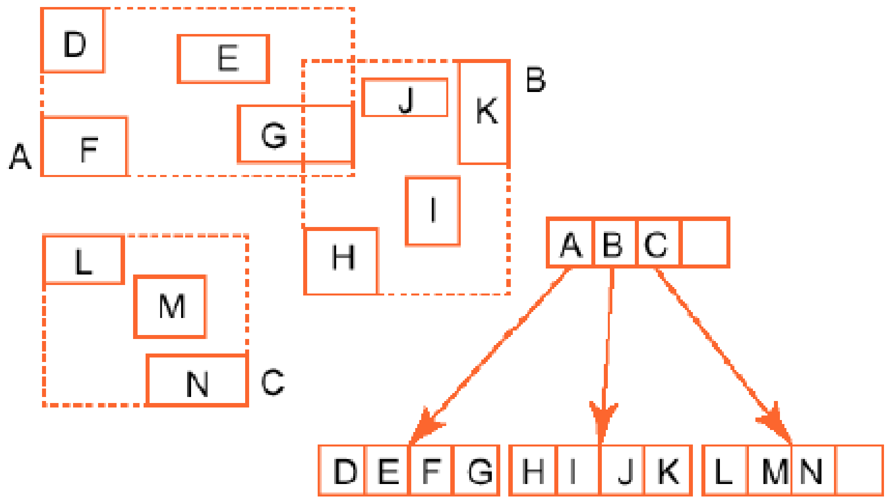


Region Quadtree

Spatial Index

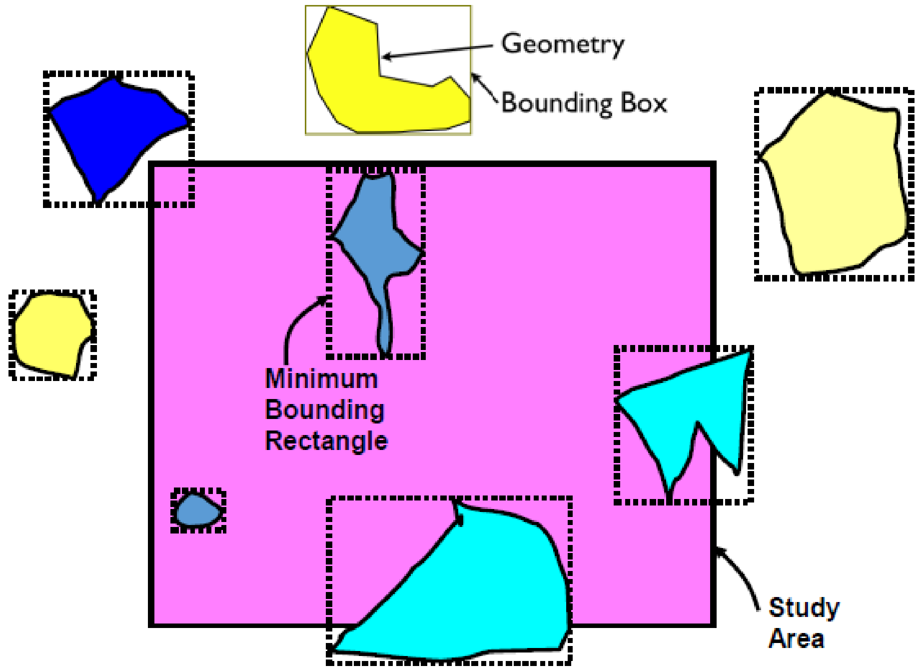
- R-tree

Use minimum bounding rectangle (MBR) or minimum bounding box (MBB)



Branching factor M = 4

- R-tree - Minimum Bounding Rectangle



Add a new object to the MBR that would expand the least to accommodate the object

PostGIS

- PostGIS is an open-source software program that adds support for geographic objects to the PostgreSQL.
- PostGIS is a spatial extension for PostgreSQL
- PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC).

- Coordinate transformation
- Identify (SRID)
- Buffer
- Touches
- Crosses
- Within
- Overlaps
- Contains

- Crosses
- Overlaps
- Contains
- Area
- Length
- Point on surface
- Return geometry as SVG

- PostGIS supports a geometry type which is compliant with the OGC standard for Simple Features.
  - POINT(50 100)
  - LINESTRING (10 10, 20 20)
  - POLYGON ((0 0, 5 5, 5 0, 0 0))
  - MULTIPOINT ((1 1), (0 0))
  - MULTILINESTRING (...)
  - MULTIPOLYGON (...)

PostGIS

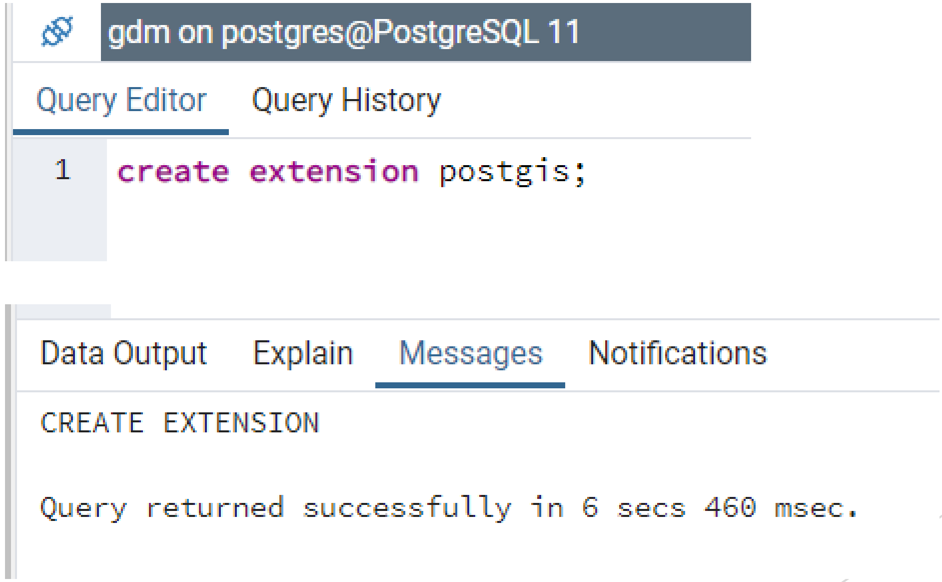
Creating a Database

- Name: gdm
- Owner: postgres

[Link For lab data : Lab Data]

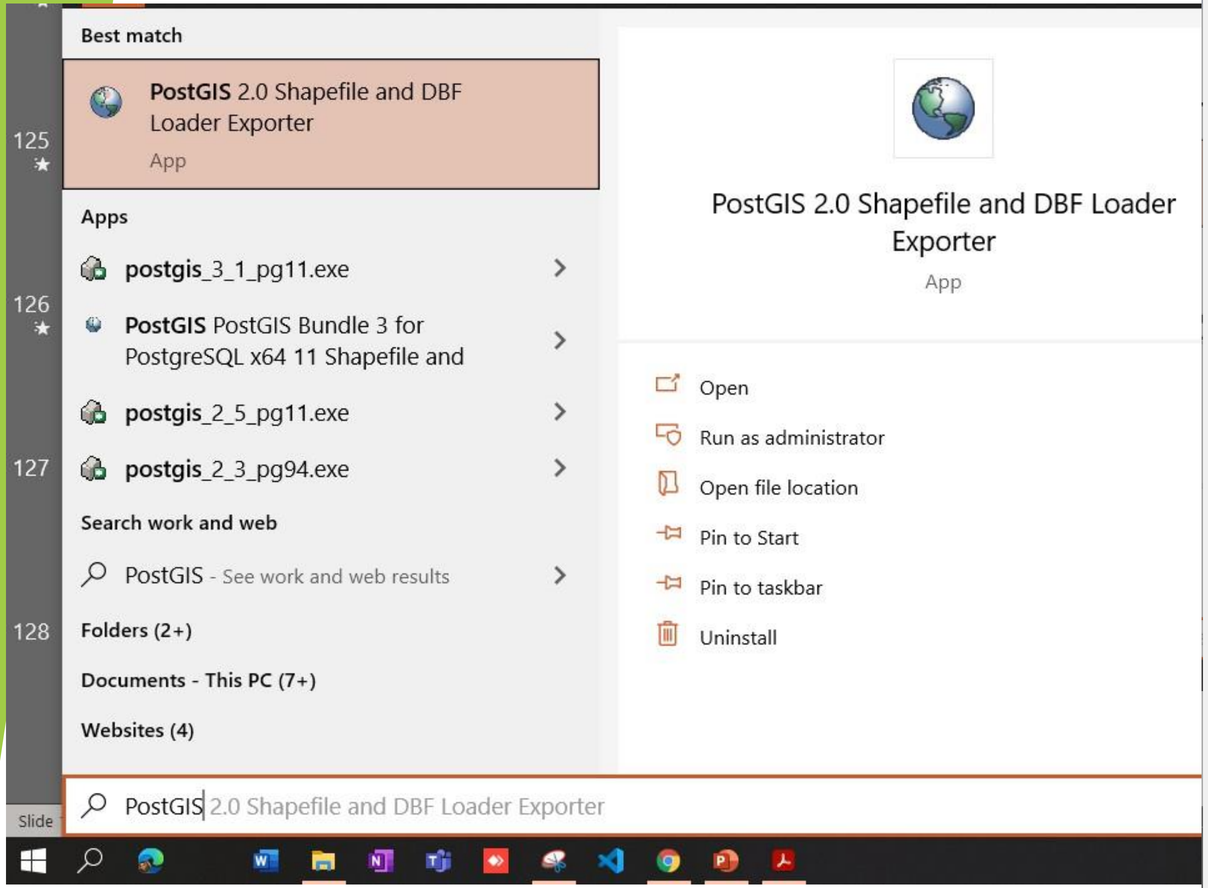
If fail, it mean you did not install PostGIS plugin yet

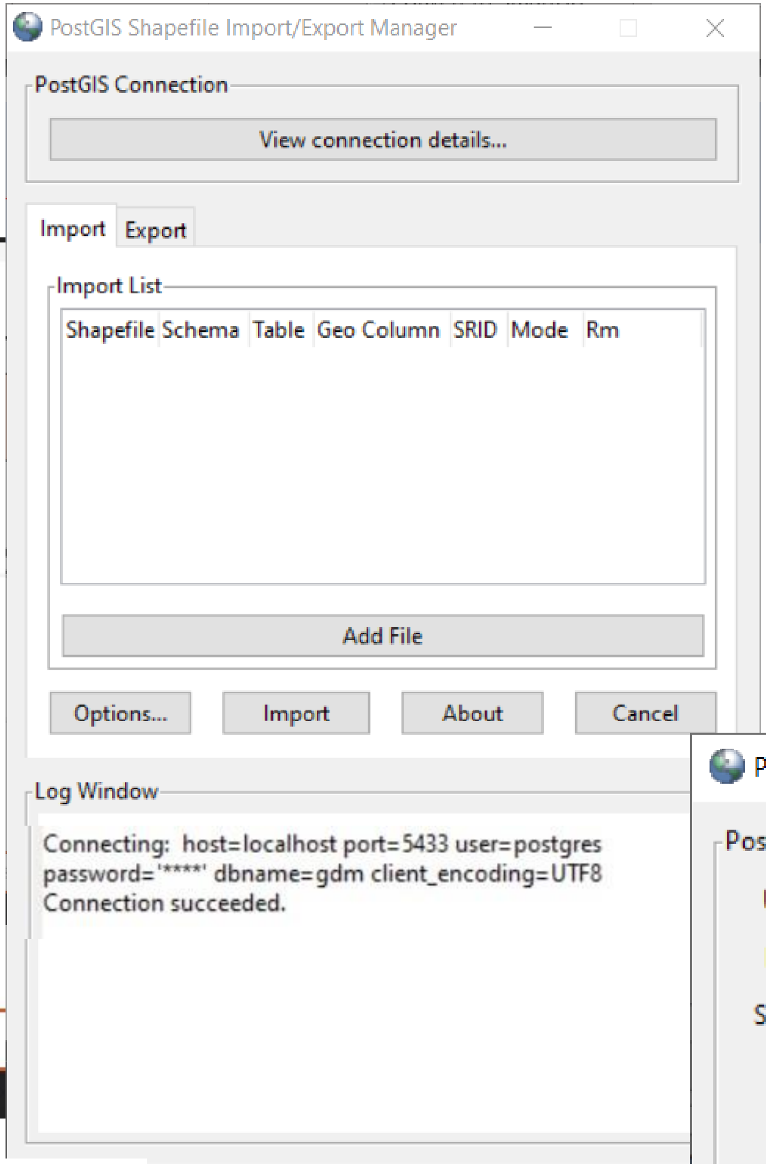
- Go to Tools, Query Tool

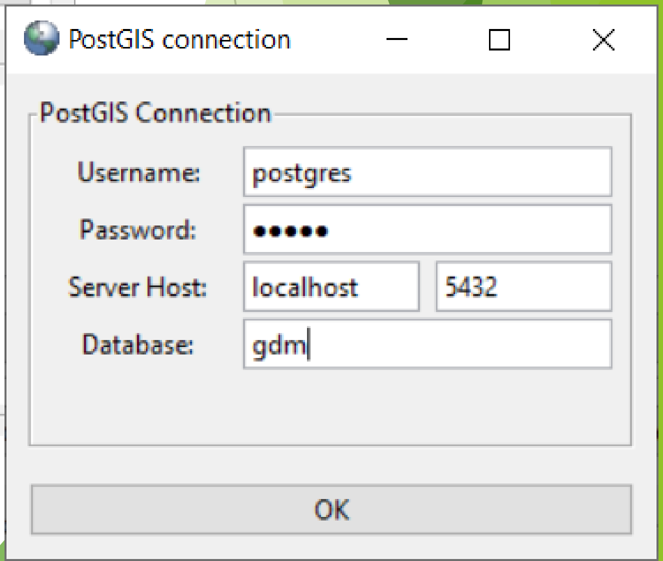


PostGIS

Loading spatial data

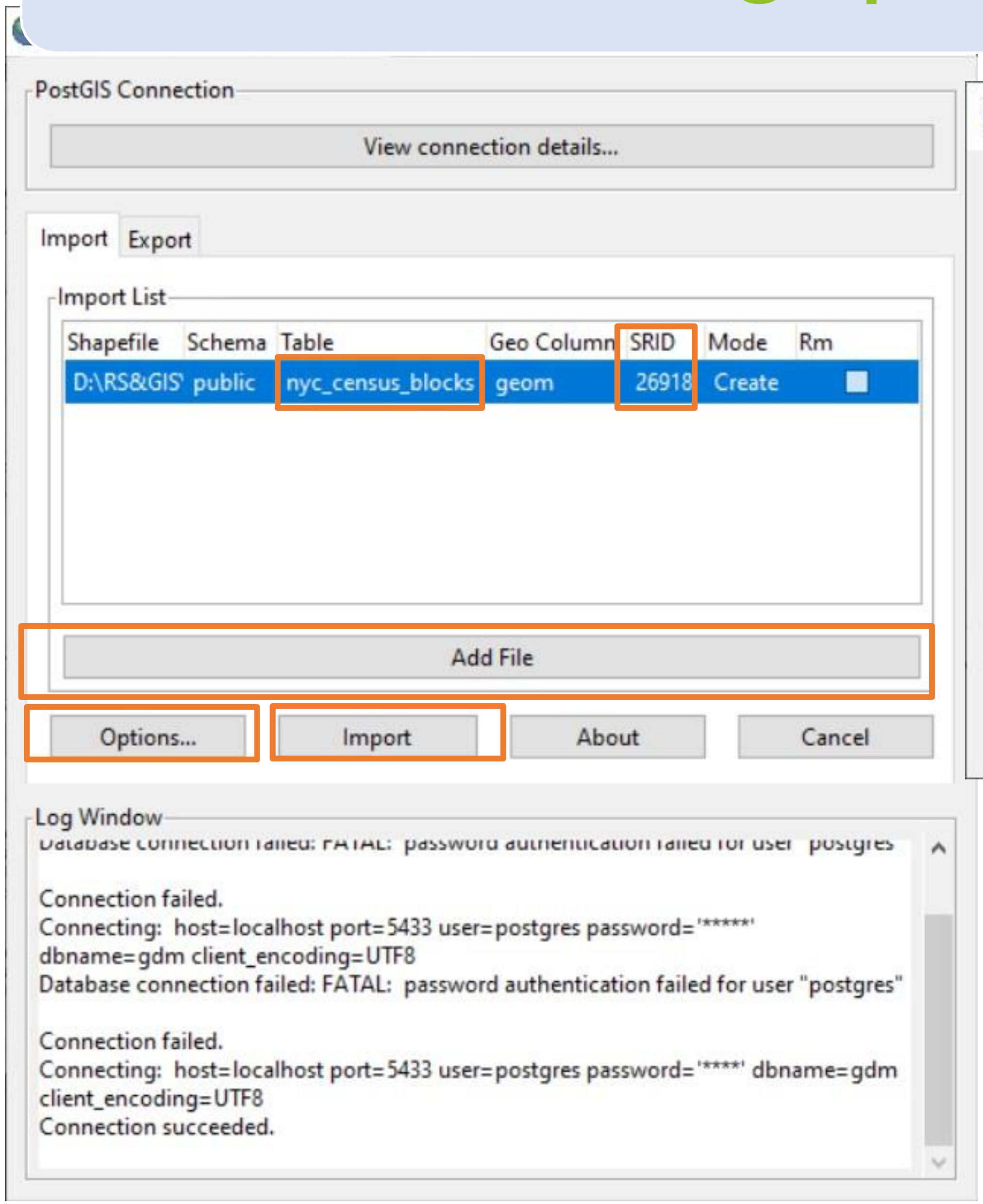


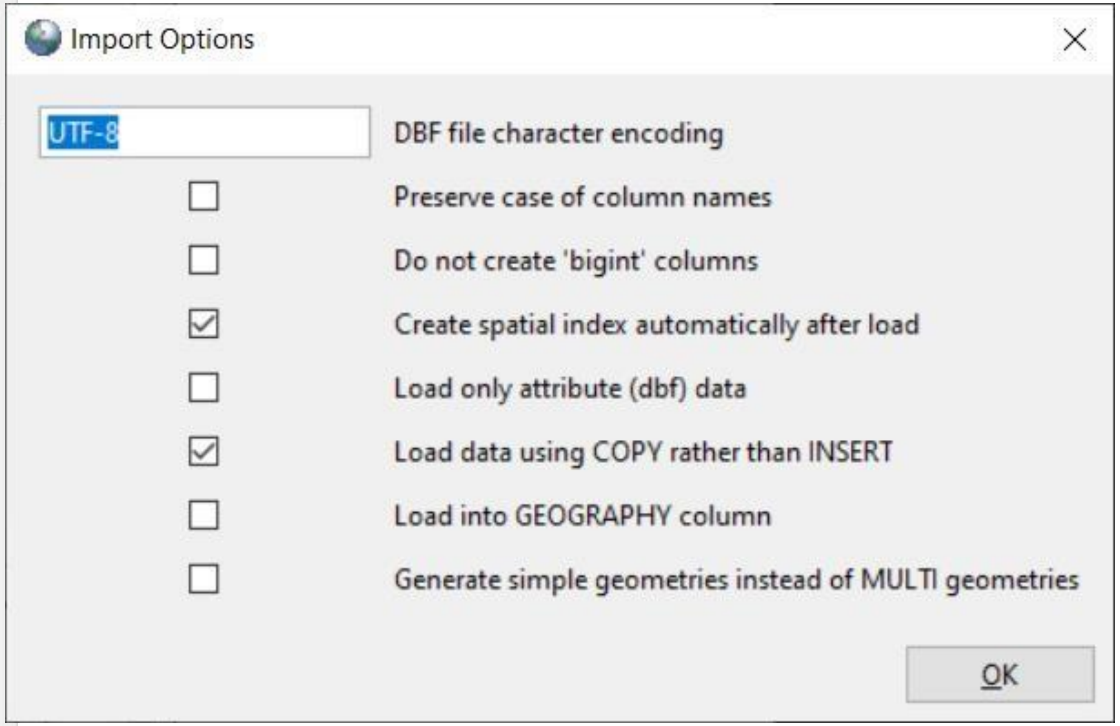





```
alter user postgres with password 'root';
```

PostGIS-Loading spatial data



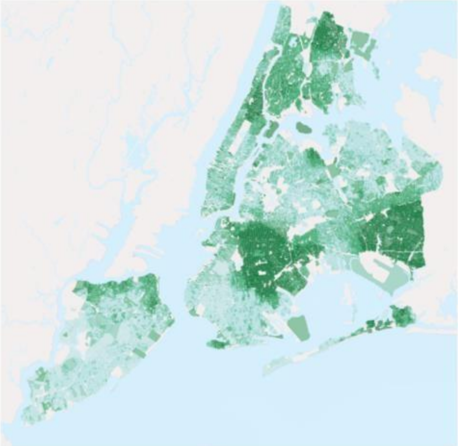


Repeat import for another 4 shp

- nyc_streets.shp
- nyc_neighborhoods.shp
- nyc_subway_stations.shp
- nyc_homicides.shp

Data Details

- nyc_census_blocks: A census block is the smallest geography for which census data is reported. All higher level census geographies (block groups, tracts, metro areas, counties, etc) can be built from unions of census blocks.



blkid	A 15-digit code that uniquely identifies every census **block**. Eg: 360050001009000
popn_total	Total number of people in the census block
popn_white	Number of people self-identifying as “White” in the block
popn_black	Number of people self-identifying as “Black” in the block
popn_nativ	Number of people self-identifying as “Native American” in the block
popn_asian	Number of people self-identifying as “Asian” in the block
popn_other	Number of people self-identifying with other categories in the block
boroname	Name of the New York borough. Manhattan, The Bronx, Brooklyn, Staten Island, Queens
geom	Polygon boundary of the block

- nyc_neighborhoods: Neighborhoods are social constructs that do not follow lines laid down by the government. For example, the Brooklyn neighborhoods of Carroll Gardens, Red Hook, and Cobble Hill were once collectively known as “South Brooklyn.

name	Name of the neighborhood
boroname	Name of the New York borough. Manhattan, The Bronx, Brooklyn, Staten Island, Queens
geom	Polygon boundary of the neighborhood

Data Details

- nyc_streets: The street centerlines form the transportation network of the city

name	Name of the street
oneway	Is the street one-way? "yes" = yes, "" = no
type	Road type (primary, secondary, residential, motorway)
geom	Linear centerline of the street

- nyc_subway_stations: subway station in new york

name	Name of the station
borough	Name of the New York borough. Manhattan, The Bronx, Brooklyn, Staten Island, Queens
routes	Subway lines that run through this station
transfers	Lines you can transfer to via this station
express	Stations where express trains stop, "express" = yes, "" = no
geom	Point location of the station

Data Details

- nyc_census_sociodata: There is a rich collection of social-economic data collected during the census process

tractid	An 11-digit code that uniquely identifies every census **tract**. ("36005000100")
transit_total	Number of workers in the tract
transit_private	Number of workers in the tract who use private automobiles / motorcycles
transit_public	Number of workers in the tract who take public transit
transit_walk	Number of workers in the tract who walk
transit_other	Number of workers in the tract who use other forms like walking / biking
transit_none	Number of workers in the tract who work from home
transit_time_mins	Total number of minutes spent in transit by all workers in the tract (minutes)
family_count	Number of families in the tract
family_income_median	Median family income in the tract (dollars)
family_income_mean	Average family income in the tract (dollars)
family_income_aggregate	Total income of all families in the tract (dollars)
edu_total	Number of people with educational history
edu_no_highschool_dipl	Number of people with no high school diploma
edu_highschool_dipl	Number of people with high school diploma and no further education
edu_college_dipl	Number of people with college diploma and no further education
edu_graduate_dipl	Number of people with graduate school diploma

Simple SQL

1. What are the names of all the neighborhoods in New York City?
   - SELECT name FROM nyc_neighborhoods;
2. What is the number of letters in the names of all the neighborhoods in Brooklyn?
   - SELECT char_length(name) FROM nyc_neighborhoods WHERE boroname = 'Brooklyn';
3. What is the average number of letters and standard deviation of number of letters in the names of all the neighborhoods in Brooklyn?
   - SELECT avg(char_length(name)), stddev(char_length(name)) FROM nyc_neighborhoods WHERE boroname = 'Brooklyn';
4. What is the average number of letters in the names of all the neighborhoods in New York City, reported by borough?
   - SELECT boroname, avg(char_length(name)), stddev(char_length(name))
FROM nyc_neighborhoods
GROUP BY boroname;

Simple SQL

5. What is the population of the City of New York?
    - SELECT Sum(popn_total) AS population FROM nyc_census_blocks;
6. What is the population of Brooklyn?
    - SELECT sum(popn_total) AS population FROM nyc_neighborhoods

      WHERE boroname = 'Brooklyn';
7. For each borough, what percentage of the population is native?
    - SELECT boroname, 100 * Sum(popn_nativ)/Sum(popn_total) AS nativ_pct
    - FROM nyc_census_blocks
    - GROUP BY boroname;

Spatial Query

1. What is the area of the 'West Village' neighborhood?
   - SELECT ST_Area(geom) FROM nyc_neighborhoods WHERE name = 'West Village';
2. What is the area of Manhattan in acres?
   - SELECT Sum(ST_Area(geom)) / 4047 FROM nyc_neighborhoods WHERE boroname = 'Manhattan';
3. What is the total length of streets (in kilometers) in New York City?
   - SELECT Sum(ST_Length(geom)) / 1000 FROM nyc_streets;
4. What is the JSON representation of the boundary of the 'West Village'?
   - SELECT ST_AsGeoJSON(geom) FROM nyc_neighborhoods WHERE name = 'West Village';
5. What is the length of streets in New York City, summarized by type
   - SELECT type, Sum(ST_Length(geom)) AS length FROM nyc_streets GROUP BY type ORDER BY length DESC;

Spatial Joins

- What is the population of the neighborhoods of Manhattan?

```
SELECT  neighborhoods.name AS neighborhood_name,
  Sum(census.popn_total) AS population
FROM nyc_neighborhoods AS neighborhoods
JOIN nyc_census_blocks AS census
ON ST_Intersects(neighborhoods.geom, census.geom)
WHERE neighborhoods.boroname = 'Manhattan'
GROUP BY neighborhoods.name
ORDER BY population DESC;
```

- What are the population density (people / km^2) of the 'Upper West Side' and 'Upper East Side'?

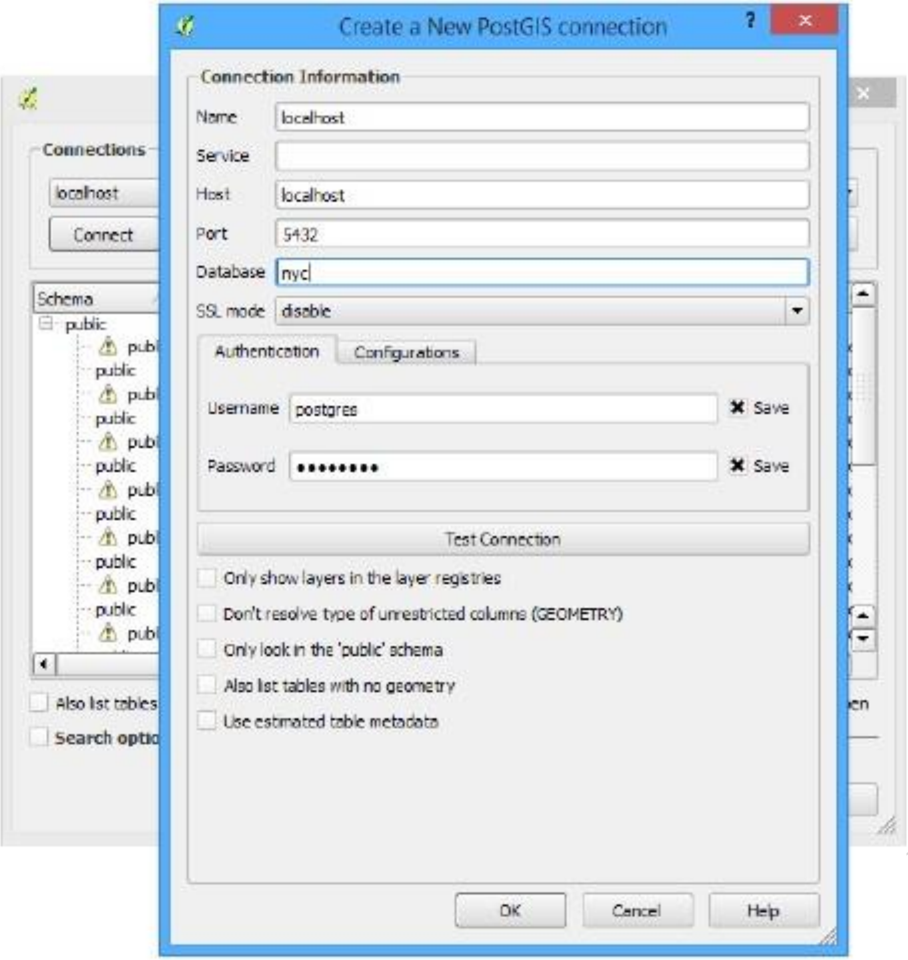
```
SELECT n.name,
  Sum(c.popn_total) / (ST_Area(n.geom) / 1000000.0) AS popn_per_sqkm
FROM nyc_census_blocks AS c
JOIN nyc_neighborhoods AS n ON ST_Intersects(c.geom, n.geom)
WHERE n.name = 'Upper West Side' OR n.name = 'Upper East Side'
GROUP BY n.name, n.geom;
```

Spatial Indexing

- Try drop out index
  - DROP INDEX nyc_census_blocks_geom_gist;
- Try select something
  - SELECT blocks.blkid FROM nyc_census_blocks blocks
    JOIN nyc_subway_stations subways
    ON ST_Contains(blocks.geom, subways.geom)
    WHERE subways.name = 'Broad St';
  - 32ms (on my pc)
- Try create index
  - CREATE INDEX nyc_census_blocks_geom_idx
    ON nyc_census_blocks
    USING GIST (geom);
  - 12ms (on my pc)

Access Shape from QGIS

- Open QGIS
- Add Layer
- Add PostGIS Layer



Access Shape from uDig

- Open uDig
- Layer -> Add, Select PostGIS