Geodatabase and web-GIS as decision support system for precision agriculture: Theory and Application

=Part I Geodatabase=

SARAWUT NINSAWAT
REMOTE SENSING AND GIS FOS
ASIAN INSTITUTE OF TECHNOLOGY

Introduction

The amount of information available to us is literally exploding and the value of data as an organizational asset is widely recognized.

Users require tools that simplify the tasks of managing the data and extracting useful information in a timely fashion.

Otherwise, data can become a liability, with the cost of acquiring it and managing it far exceeding the value derived from it.

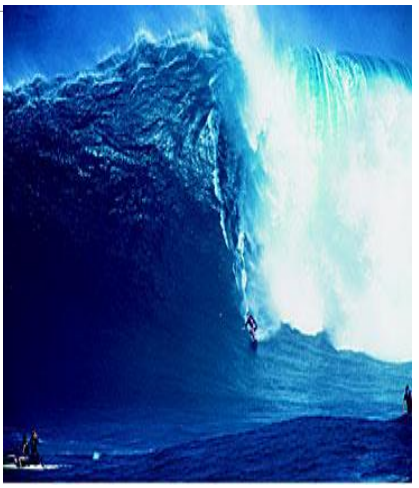
Trends leading to Data Flood

More data is generated:

- Bank, telecom, other business transactions ...
- Scientific Data: astronomy, biology, etc
- Web, text, and e-commerce

More data is captured:

- Storage technology faster and cheaper
- DBMS capable of handling bigger DB



3

Examples

Europe's Very Long Baseline Interferometry (VLBI) has 16 telescopes, each of which produces **1 Gigabit/second** of astronomical data over a 25-day observation session

- storage and analysis a big problem

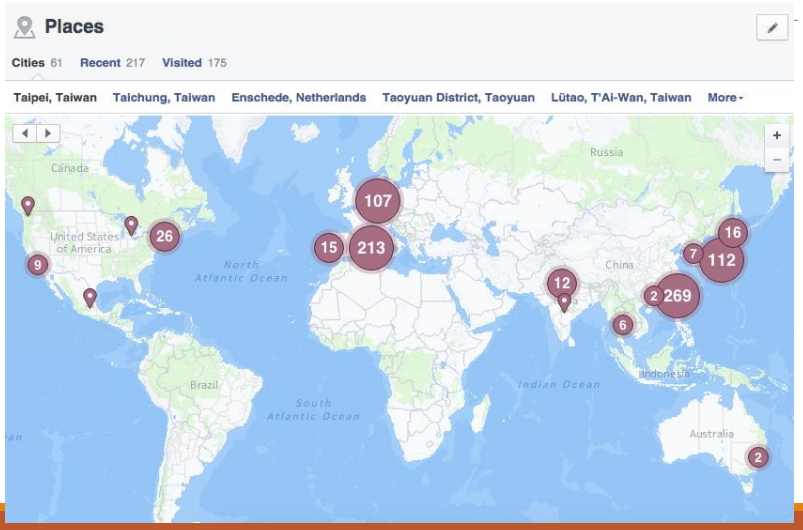
Walmart reported to have 24 Tera-byte DB

AT&T handles billions of calls per day

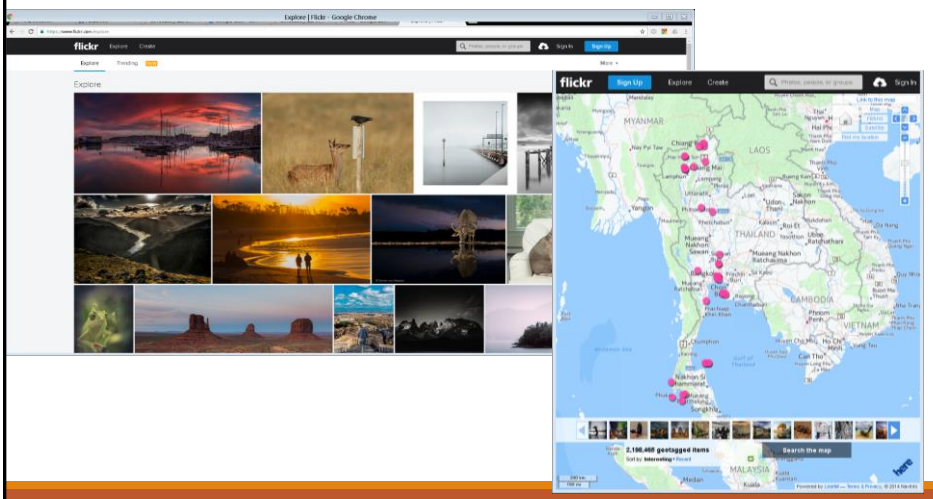
- data cannot be stored -- analysis is done on the fly

4

Do you check in today ?



Do you share photo on the web ?



Do you twitter what you see?

Courtesy from http://www.businessinsider.com/incredible-things-that-happen-every-60-seconds-on-the-internet-2011-12

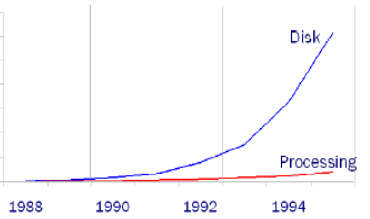
Growth Trends

Moore's law
- Computer Speed doubles every 18 months

Storage law
- total storage doubles every 9 months

Consequence
- very little data will ever be looked at by a human

Knowledge Discovery is **NEEDED** to make sense and use of data.



9

Data vs. Information

Data	Information
raw facts	data with context
no context	processed data
just numbers and text	value-added to data ◦ summarized ◦ organized ◦ analyzed

Data vs. Information

Data: 51007

Information:

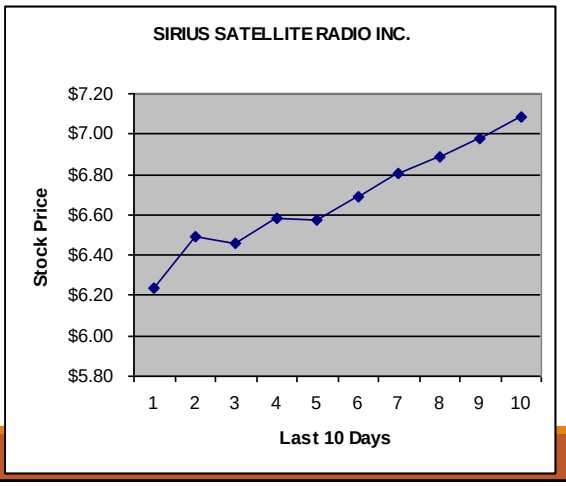
- 5/10/07 The date of your final exam.
- $51,007 The average starting salary of an accounting major.
- 51007 Zip code of Bronson Iowa.

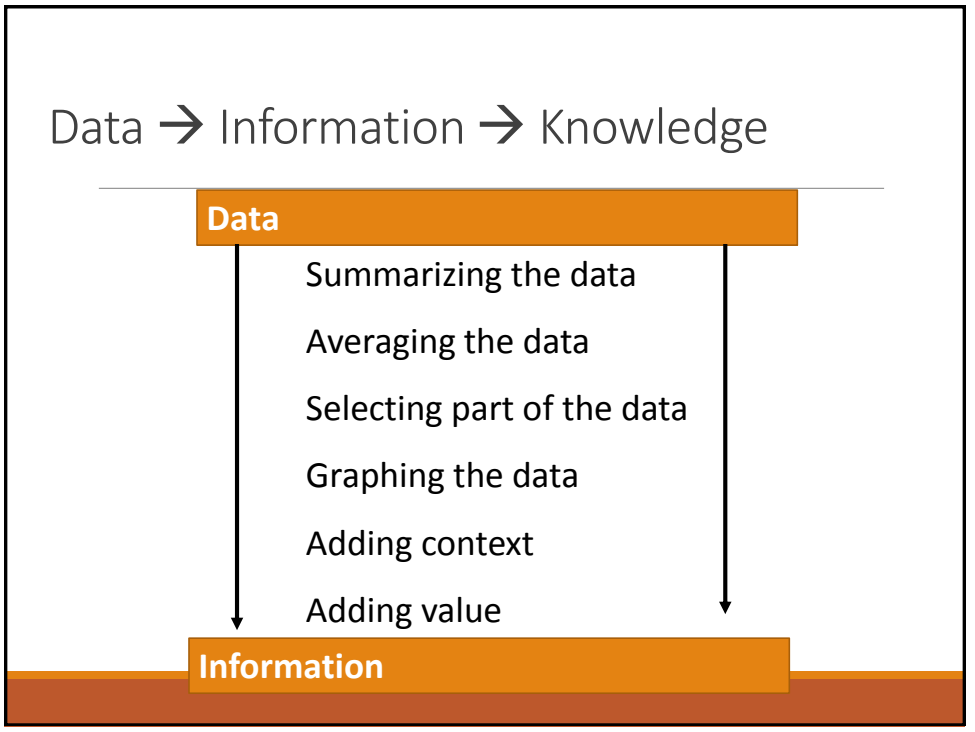
Data vs. Information

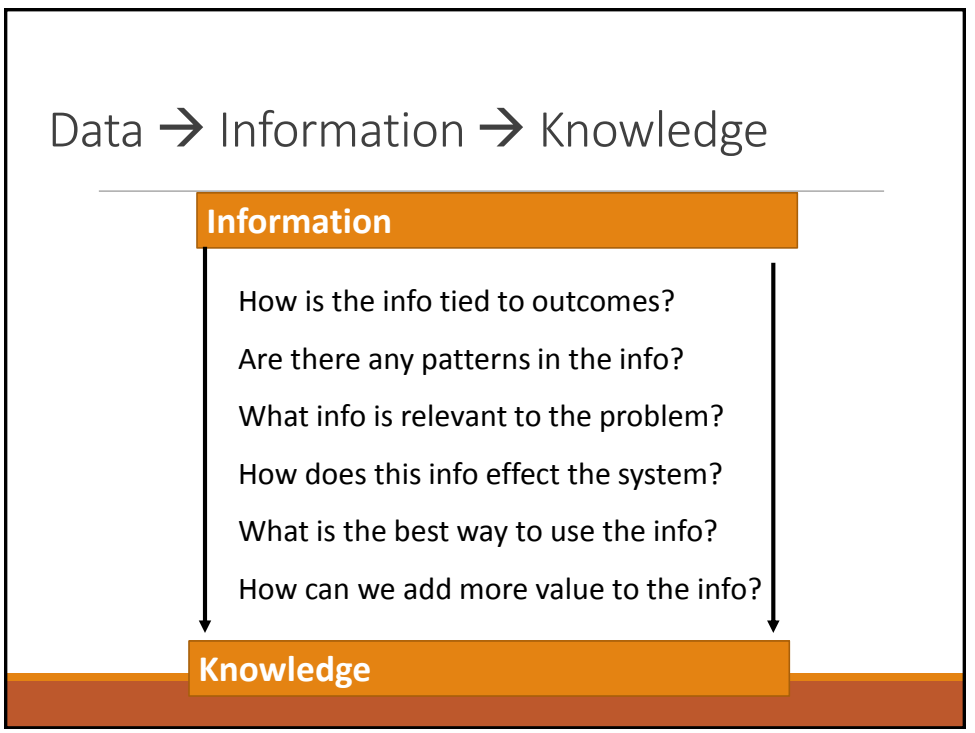
Data

6.34
6.45
6.39
6.62
6.57
6.64
6.71
6.82
7.12
7.06

Information

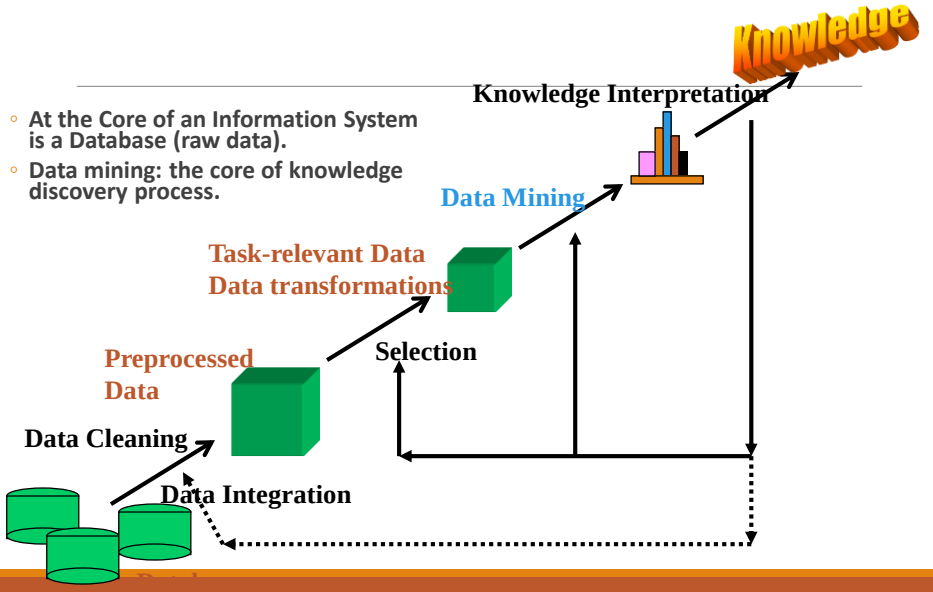



Data → Information → Knowledge
Data
Summarizing the data
Averaging the data
Selecting part of the data
Graphing the data
Adding context
Adding value
Information


Data → Information → Knowledge
Information
How is the info tied to outcomes?
Are there any patterns in the info?
What info is relevant to the problem?
How does this info effect the system?
What is the best way to use the info?
How can we add more value to the info?
Knowledge

Knowledge Discovery Process

- At the Core of an Information System is a Database (raw data).
- Data mining: the core of knowledge discovery process.

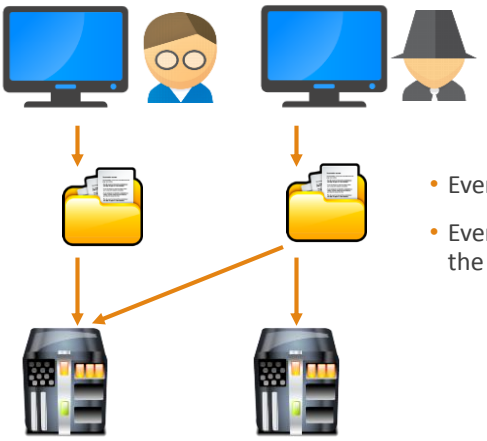


Database

A database is a collection of data, typically describing the activities of one or more related organizations. For example, a university database might contain information about the following :

- Entities - Students, Faculty, coursed and classrooms
- Relationships between entities such as student's enrollment in course, faculty teaching course, and the use of rooms for courses.

Record Files



- Every application has it own data
- Every application keeps its data in the files that it manages itself.

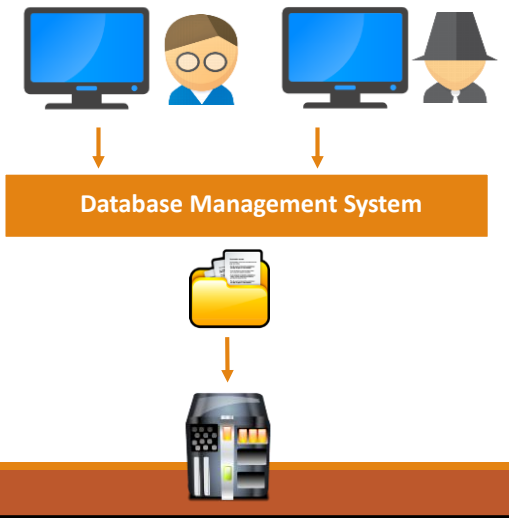
File based data disadvantages

- Redundancy - Same data is kept in multiple places
  - Waste of disk space
- Inconsistency - Multiple copies of same data can become different
  - Inconsistency
  - Versioning
- Loss of Integrity - It is difficult to keep the data correct
- Difficult in new application - A lot of work must be duplicated for every new application
- Policy gaps
- Security

Examples

- The names, numbers and departments of students are kept both in Student Affairs and in the Library
- The name of the same student can be recorded
  - as ”Victoria Adams”in Student Affairs and
  - as ”Victoria Beckham”in the Library
- The ”Control and Computer Engineering” department is closed but the department data of its students remains the same
- A new application will be developed for the Scholarship Office.
- Each department considers only its own requirements
- Security depends only on the operating system

Database Management Systems



- Data are kept in a share system.
- Application access data over a common interface.

Why put spatial data in a RDBMS?

- Spatial data is usually related to other type of complex spatial data. Allows one to encode more relationships.
  - Fire Hydrant: number of uses, service area, last maintenance date.
  - River: flow, temperature, fish presence, chemical concentrations
  - Forested Area: monetary value, types of trees, ownership

Historically?

- In early GIS implementations, spatial data and related attribute information were stored separately. The attribute information was in a database (or flat file), while the spatial information was in a separate, proprietary, GIS file structure.

  For example, municipalities often would store property line information in a GIS file and ownership information in a database.

- Spatial databases were born when people started to treat spatial information as first class database objects.

What is a Spatial Database?

Database that:

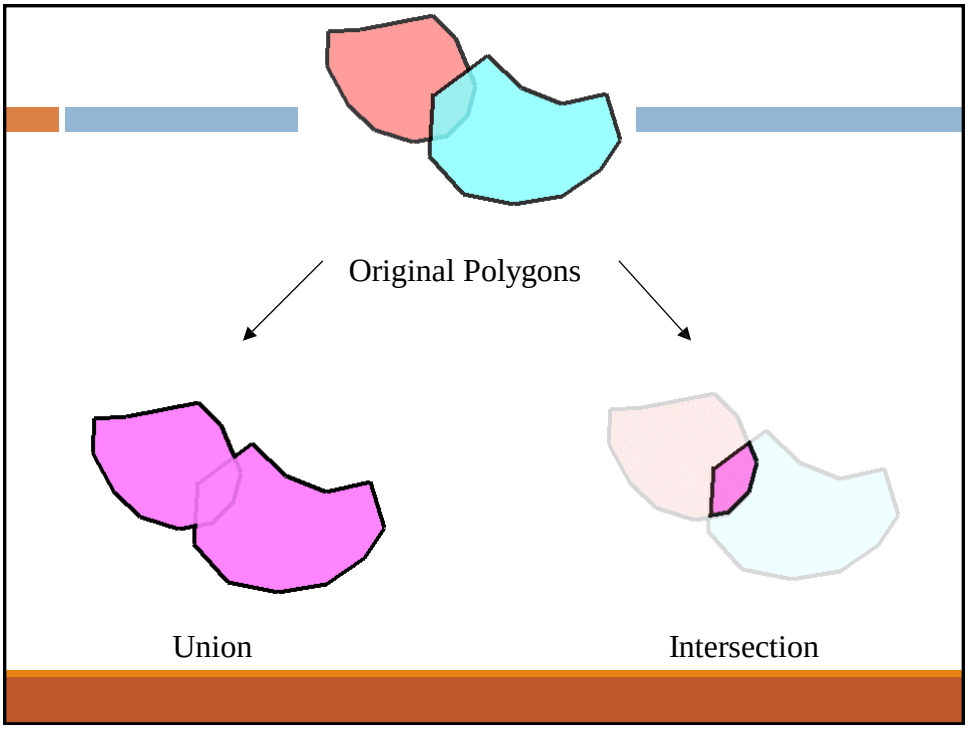
- Stores spatial objects
- Manipulates spatial objects just like other objects in the database

Why Spatial Database?

- Database do **some things** better than files.
  - Support multiple users editing and accessing the same data at the same time.
  - Support large data volumes better than files. (For some kinds of data.)
  - Provide a unified means of accessing and analysing data using **SQL abstraction.**
  - Provide a unified means of access control for data.
  - Provide an integration point for spatial data and enterprise data (usual DB).

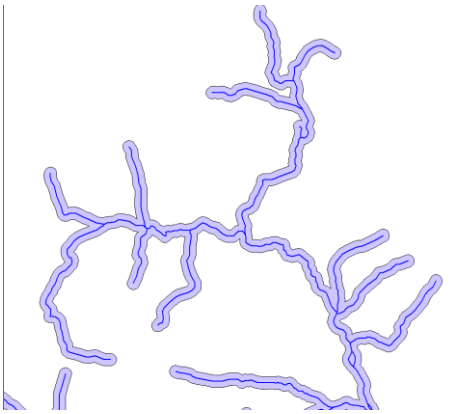
Spatial Relationships

- Not just interested in location, also interested in "Relationships" between objects that are very hard to model outside the spatial domain.
- The most common relationships are
  - Proximity : distance
  - Adjacency : "touching" and "connectivity"
  - Containment : inside/overlapping



Spatial Relationships

Original river network



Buffered rivers

Spatial relationship in Facebook Functions

… WHERE distance(<me>,pub_loc) < 1000

SELECT distance(<me>,pub_loc)*$0.01 + beer_cost …

… WHERE touches(pub_loc, street)

… WHERE inside(pub_loc,city_area) and city_name = …

Spatial Relationships query

```
select name, beer_price,
      distance(location, GeometryFromText('POINT(1195722
383854)',2167)) from pubs order by beer_price;
```

name	beer_price	distance
Fireside	4.25	1484.10275160491
The Forge	4.33	1533.06561109862
Rumours	4.46	2042.00094093097
Garricks Head	4.5	669.389105609889
Slap Happy	4.5	1882.31910168298
Old Bailys	4.55	1147.20900404641
Black Sheep	4.66	536.859935972633
Big Bad Daves	4.75	907.446543878884

Spatial Relationships query

```
select name, beer_price + 0.001 * distance(location,
GeometryFromText('POINT(1195722 383854)',2167)) as net_price
                  from pubs order by price;

     name       |     net_price
----------------+-------------------
 Garricks Head | 5.16938910560989
 Black Sheep   | 5.19685978338474
 Big Bad Daves | 5.65744654387888
 Old Bailys    | 5.69720919478127
 Fireside      | 5.73410275160491
 The Forge     | 5.86306553480468
 Slap Happy    | 6.38231910168298
 Rumours       | 6.50200097907794
```

Disadvantages of Spatial Databases

- Cost to implement can be high
- Some inflexibility
- Incompatibilities with some GIS software
- Slower than local, specialized data structures
- User/managerial inexperience and caution

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
- SpatialLite
- PostGIS / PostgreSQL

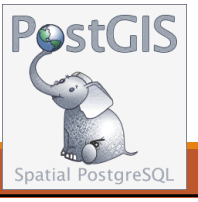
PostgreSQL

18

- PostgreSQL is a powerful, object-relational database management system (ORDBMS).
- It is released under a BSD-style license and is thus free and open source software.
- As with many other open source programs, PostgreSQL is not controlled by any single company, but has a global community of developers and companies to develop it.

PostGIS

- *PostGIS* is a spatial extension for PostgreSQL
  - Enable PostgreSQL Database Management System into a spatial database by adding adding support for the three features:
    - Spatial types, Indexes and Functions.
- *PostGIS* aims to be an "OpenGIS Simple Features for SQL" compliant spatial database



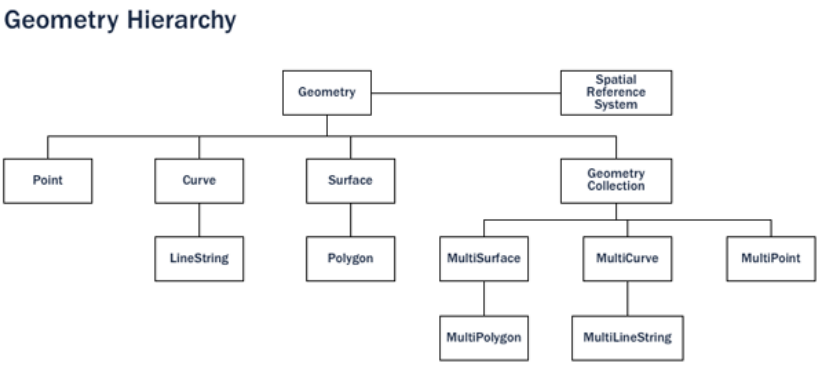
Spatial Type

20

- An ordinary database has strings, numbers, and dates. A spatial database adds additional "spatial" types for representing **geographic features**.
- Spatial data types are organized in a type hierarchy. Each sub-type inherits the structure (attributes) and the behavior (methods or functions) of its super-type.

Geometry Hierarchy

21



Point/Multipoint Geometry

22

- POINT(0 0)
- MULTIPOINT(0 1,1 0,2 1,1 2)

Linestring/Multilinestring Geometry

23

- LINESTRING(1 1,2 0,3 1,3 3,2 4)
- MULTILINESTRING((0 2,1 3,2 2),(1 1,2 0,3 1,3 3,2 2))

Simple non-closed linestring

Simple multilinestring defined by 4 endpoints of 2 elements

Polygon/Multipolygon Geometry

24

- POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
- MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))

Geometrycollection

25

- GEOMETRYCOLLECTION(POINT(2 0),POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))

3D Geometries

26

- PostGIS recognizes and stores 3D geometries, but not yet full support.
- Lack the volumetric sense of 3D
  - 2D object sitting in 3D space
  - 2.5D

Spatial Indexing

Used the GiST (Generalized Search Tree) index

- Actively being developed
  - Teodor Sigaev and Oleg Bartunov
  - http://www.sai.msu.su/~megera/postgres/gist/
- Fast index creation
- Handles compression
  - use bounding box of the feature
- NULL safe
- Can implement an R-Tree using GiST

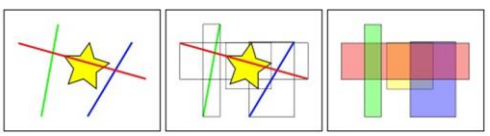
Spatial Indexing

Number of lines that intersect the yellow star is one (red line)

But the bounding boxes of features that intersect the yellow box is two (red and blue)

Using spatial index created based on the geometry column; it is determined "what boxes intersect the yellow box"

Then exact calculation of "what lines intersect the yellow star" is executed only for those features returned by the first test.



44

R-Tree Indexing

- Generalize all the geometries to their bounding box.
  - small to store
  - operations are simple
- Typical search is to find all the objects that overlap a box
- Result is an approximation
  - too many features are returned
- Used to solve overlap and distance problems

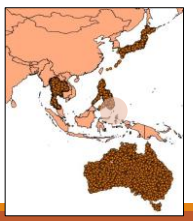
Performance Analysis with Intersection

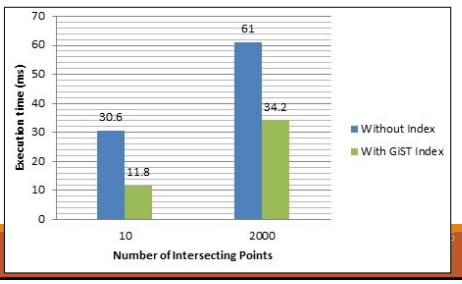
Performance was checked when earthquake geometry and location geometries intersect

Observed the execution time by increasing the number of intersecting location geometries

Total number of records were maintained at 50,000 mark

The percentage of increased execution time found to be less in the GiST-indexed table compared to the non-indexed table





Spatial Functions

29

- A spatial database provides a complete set of functions for analyzing geometric components, determining spatial relationships, and manipulating geometries.
- These spatial functions serve as the building block for any spatial project.

Spatial Functions

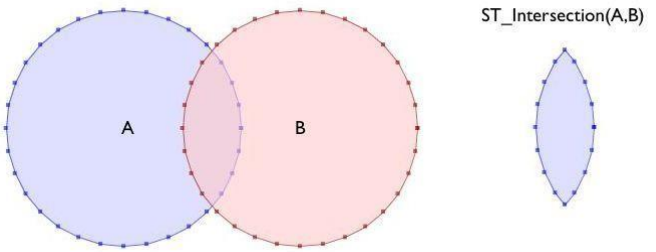
30

- **Conversion**: Functions that *convert* between geometries and external data formats.
- **Management**: Functions that *manage* information about spatial tables and PostGIS administration.
- **Retrieval**: Functions that *retrieve* properties and measurements of a Geometry.
- **Comparison**: Functions that *compare* two geometries with respect to their spatial relation.
- **Generation**: Functions that *generate* new geometries from others.

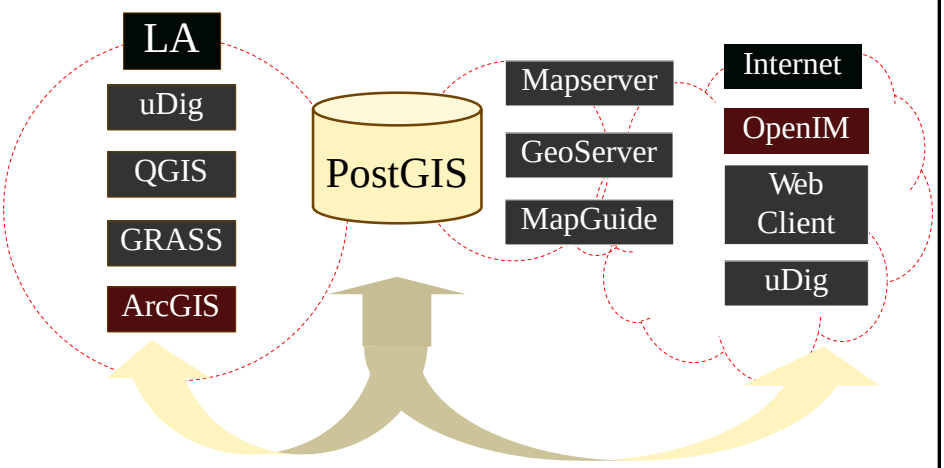
Spatial Function

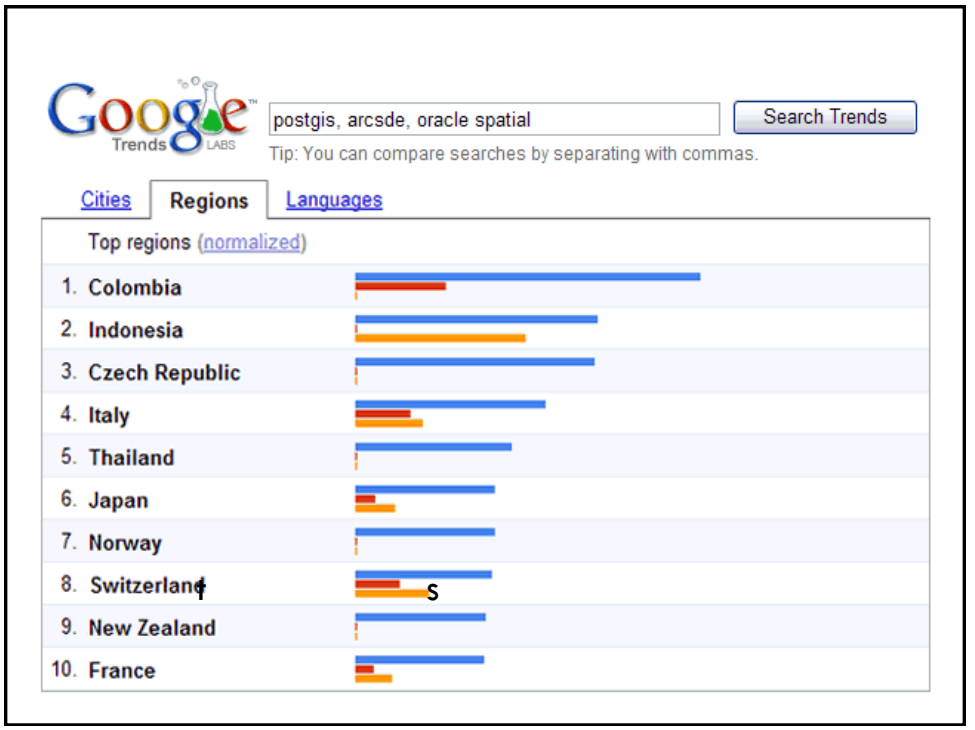
31

- SELECT ST_AsText(ST_Intersection(ST_Buffer('POINT(0 0)', 2), ST_Buffer('POINT(3 0)', 2)));



PostGIS in the Spatial Stack





Who is using PostGIS?

- The Ministry of Sustainable Resource Management (British Columbia, Canada) uses PostGIS to store, manage and analyze their Digital Road Atlas, a large and complex road network database.

Who is using PostGIS?

- Institut Géographique National (IGN France) uses PostGIS to underpin a system containing more than 100 million spatial objects, including a 3D national database of roads, rails, hydrography, vegetation, buildings and administrative boundaries.

Who is using PostGIS?

- InfoTerra (United Kingdom) uses PostGIS to store >600 million features of topographic data. Sits at the back-end of their online DataStore. Notable load times of 12 hours for the entire data-suite ~14000 features per second. More notable savings of ~£1000000 per annum.

PostgreSQL Installation

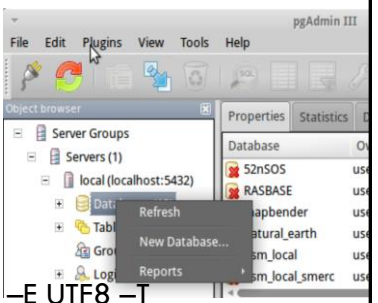
- Free/Open source (http://www.postgresql.org)
- Cross platform
- PL/PgSQL language is required for PostGIS
- Binary package
  - Linux, Mac OS X, Windows etc
  - PostgreSQL
  - PgAdmin III
- Install as Service to allow automatic database start on boot

PostGIS Installation

- Free/Open source (http://www.postgis.org/)
- Cross platform
- PL/PgSQL language is required for PostGIS
- Binary package
  - Linux, Mac OS X, Windows etc
- In Window installer, PostGIS bundled with PostgreSQL

Spatially Enable PostgreSQL

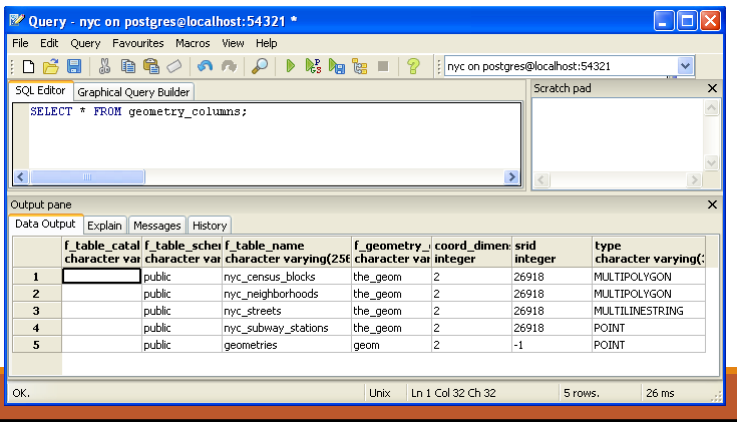
- Create a new database
- GUI in pgAdmin III
  - Select "template_postgis" as the template
  - "Template_postgis" will enable spatial function and include two tables in the database
    - spatial_ref_sys
    - geometry_columns



- Cmd:
  - createdb nyc_cmd –h localhost –U user –E UTF8 –T template_postgis
    - Remove DB : dropdb nyc_cmd –h localhost –U user

Geometry_columns table

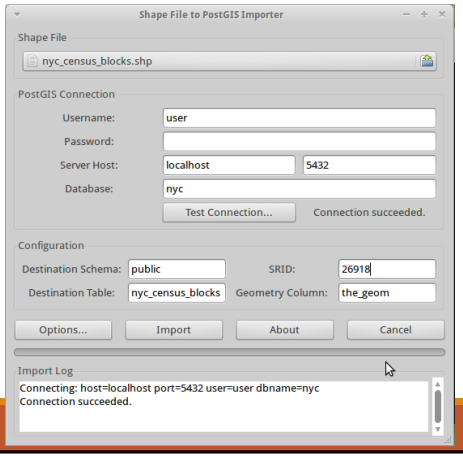
- Provides a listing of all "features" (defined as an object with geometric attributes), and the basic details of those features.



Loading Shape File

45

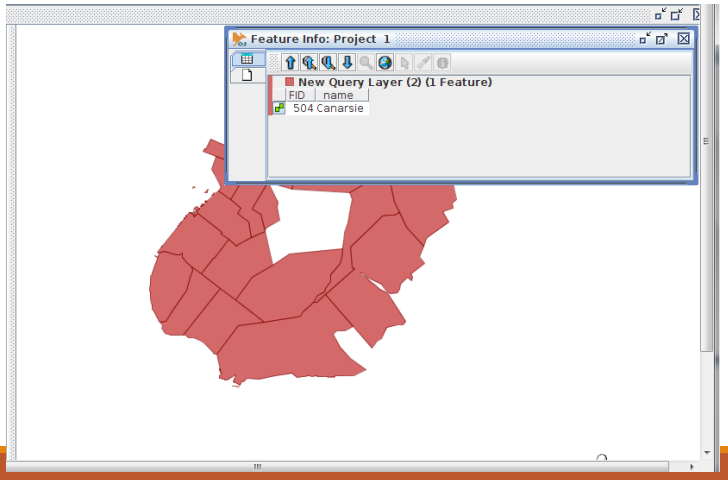
- PostGIS provides many options for loading data.
  - The GUI shapefile importer



Simple SQL Queries

52

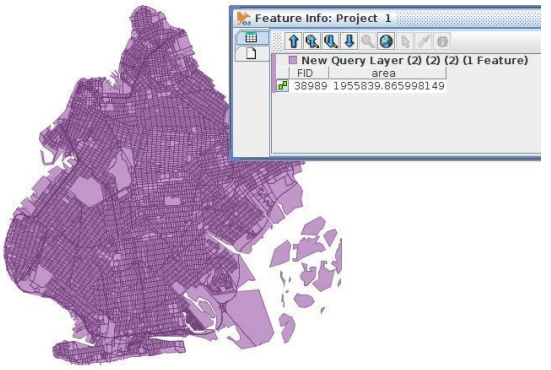
- SELECT name FROM nyc_neighborhoods WHERE boroname = 'Brooklyn';



ST_AREA function

53

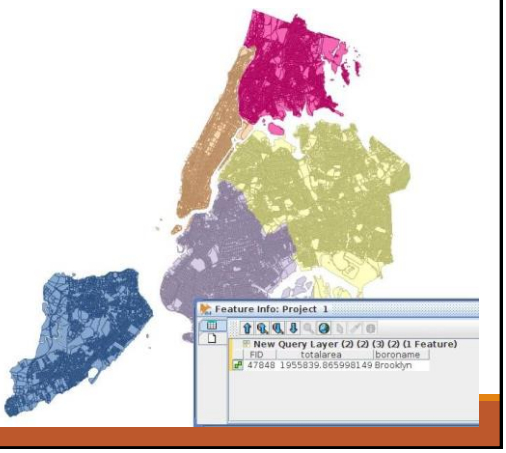
- SELECT ST_AREA(geom) as Area FROM nyc_census_blocks WHERE boroname = 'Brooklyn';



ST_AREA function

54

- SELECT SUM(ST_AREA(geom)) as TotalArea,boroname FROM nyc_census_blocks Group By boroname;



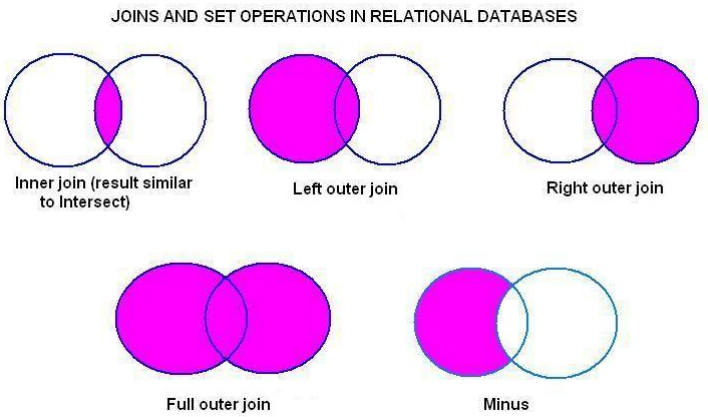
JOIN operator in RDBMS

2

- The ***JOIN*** operator specifies how to relate tables in the query.
- The JOIN operator is one of the set operations available in relational databases.
- The following join types of join are available in most relational databases:
  - INNER
  - OUTER (LEFT. RIGHT, FULL)
  - CROSS

JOIN operator

3



Result of JOIN

- INNER JOIN: Select only those rows that have values in common in the columns specified in the ON clause.
- LEFT, RIGHT, or FULL OUTER JOIN: Select all rows from the table on the left (or right, or both) regardless of whether the other table has values in common and (usually) enter NULL where data is missing.

Spatial Join

- Allow you to combine information from different tables by using spatial relationships as the join key.
- We can think that "standard GIS analysis" can be expressed as spatial joins.
- Any function that provides a true/false relationship between two tables can be used to drive a spatial join, but the most commonly used ones are:
  - **ST_Intersects, ST_Contains,** and **ST_DWithin.**
- By default, INNER JOIN is used.

ex1

6

- What is neighborhood of subway station Broad Street ?

```
SELECT subways.name AS subway_name,
  neighborhoods.name AS
  neighborhood_name,
  neighborhoods.boroname AS borough FROM
  nyc_neighborhoods AS neighborhoods
  JOIN nyc_subway_stations AS subways ON
  ST_Contains(neighborhoods.geom,
  subways.geom)      WHERE subways.name =
  'Broad St';
```

Functionally speaking

7

- ST_Contains(neighborhoods.the_geom, subways.geom) ???

neighborhoods.geom contain *subways.geom*

Ex1: result

8

```
subway_name | neighborhood_name | borough
------------+-------------------+----------
Broad St    | Financial District| Manhattan
```

Join and Summary

9

- The combination of a JOIN with a GROUP BY provides the kind of analysis that is usually done in a GIS system.
- "What is the population and racial make-up of the neighborhoods of Manhattan?"

Ex2 : Join and summary

10

```
SELECT neighborhoods.name AS neighborhood_name,
  Sum(census.popn_total) AS population, Round(100.0 *
  Sum(census.popn_white) / Sum(census.popn_total),1) AS
  white_pct, Round(100.0 * Sum(census.popn_black) /
  Sum(census.popn_total),1) AS black_pct 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
  white_pct DESC;
```

Ex2 : Result

11

```
neighborhood_name | population | white_pct | black_pct
------------------+------------+-----------+-----------
Carnegie Hill     | 19909      | 91.6      | 1.5
North Sutton Area | 21413      | 90.3      | 1.2
West Village      | 27141      | 88.1      | 2.7
Upper East Side   | 201301     | 87.8      | 2.5
Greenwich Village | 57047      | 84.1      | 3.3
Soho              | 15371      | 84.1      | 3.3
Murray Hill       | 27669      | 79.2      | 2.3
Gramercy          | 97264      | 77.8      | 5.6

…
```

Ex2 : Join and summary

12

- The **JOIN** clause creates a virtual table that includes columns from both the neighborhoods and census tables.
- The **WHERE** clause filters our virtual table to just rows in Manhattan.
- The remaining rows are grouped by the neighborhood name and fed through the aggregation function to **Sum()**the population values.
- After a little arithmetic and formatting

  (e.g., GROUP BY, ORDER BY) on the final numbers, our query spits out the percentages.

Ex3

13

- "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;
```