UNIVERSITY OF TWENTE.

Spatial Database

Presented by:

Mahdi Farnaghi

Assistant Prof.

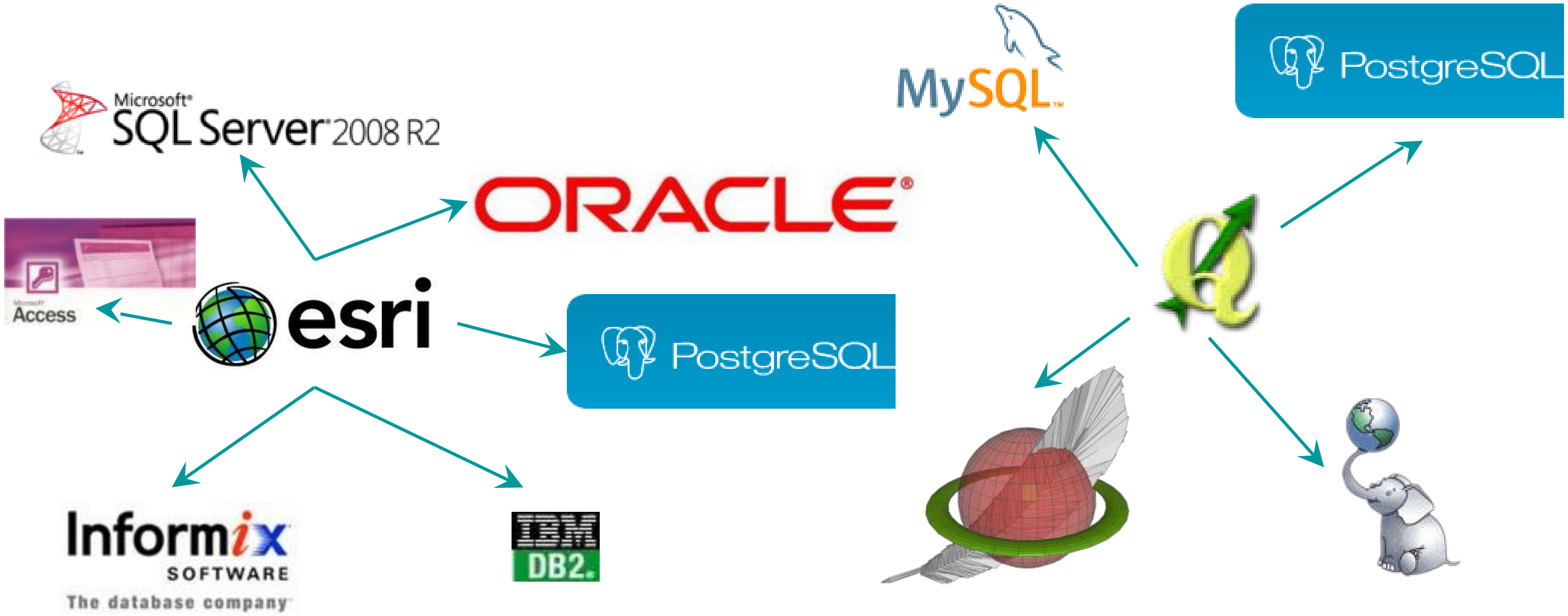
Department of Geo-Information Processing

ITC FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

DBMS AND GIS

A GIS **is not** a DBMS

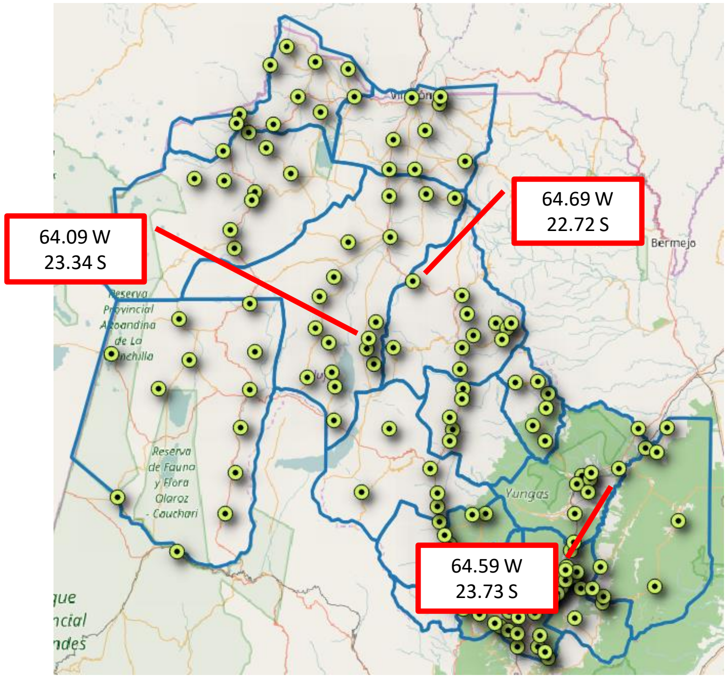
Main GIS packages **can be linked** to a DBMS to store and manage data


Microsoft SQL Server 2008 R2
ORACLE
Microsoft Access
esri
PostgreSQL
Informix SOFTWARE The database company
IBM DB2.
MySQL
PostgreSQL

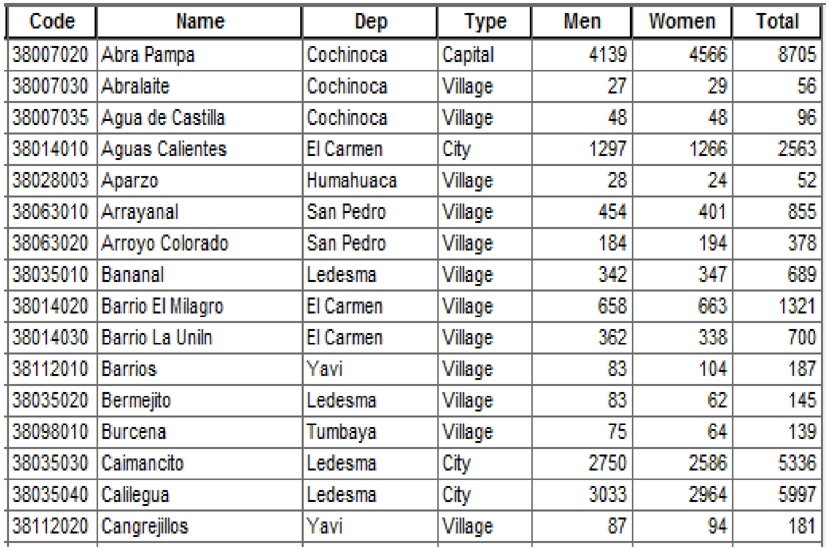
GIS AND DATABASES

LINKING SPATIAL AND NON-SPATIAL DATA

Spatial data


64.09 W
23.34 S
64.69 W
22.72 S
64.59 W
23.73 S
Bermejo
Reserva Provincial Altoandina de La Chinchilla
Reserva de Fauna y Flora Olaroz - Cauchari
Yungas

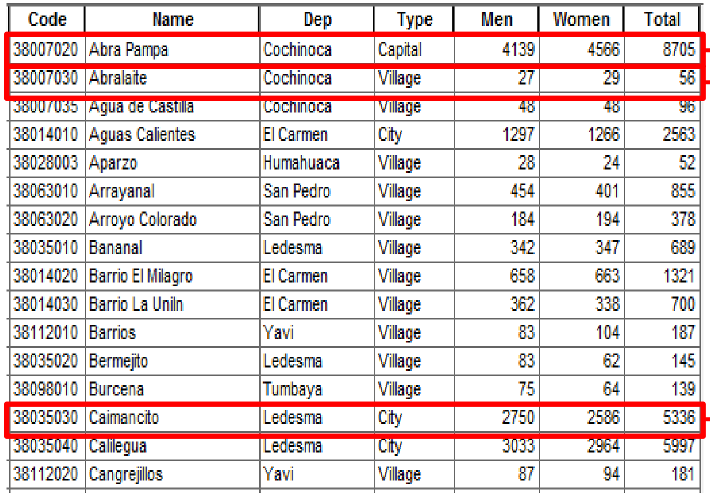
Non-spatial data



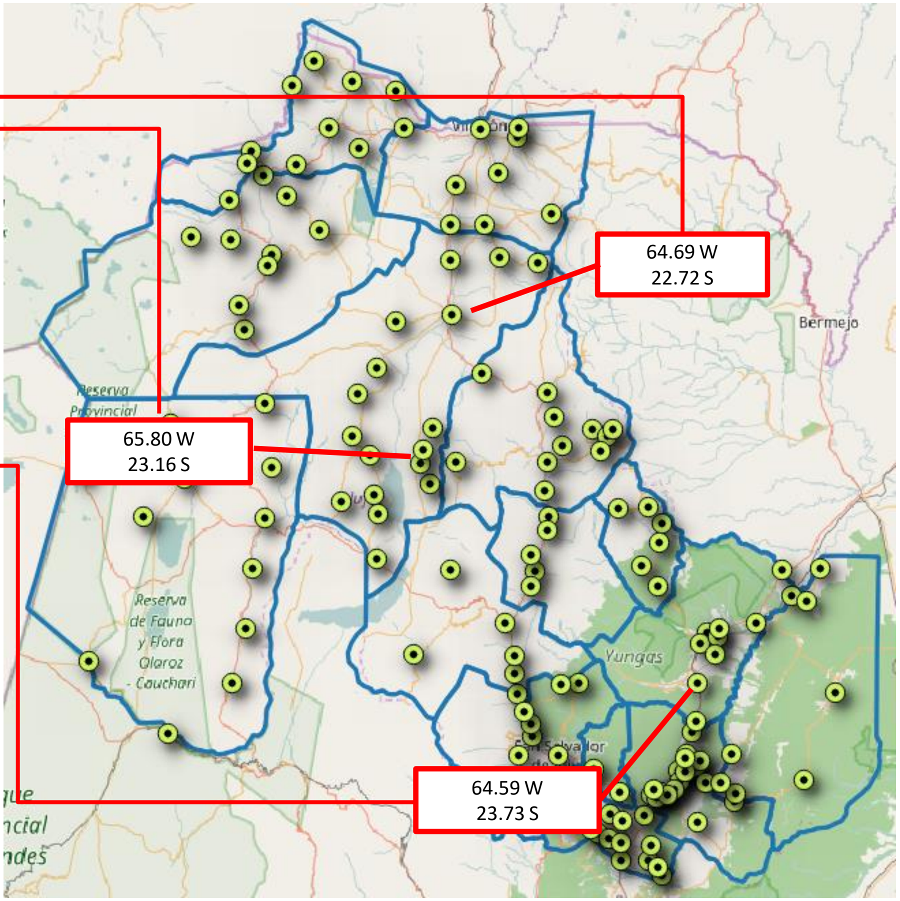
Code	Name	Dep	Type	Men	Women	Total
38007020	Abra Pampa	Cochinoca	Capital	4139	4566	8705
38007030	Abralaite	Cochinoca	Village	27	29	56
38007035	Agua de Castilla	Cochinoca	Village	48	48	96
38014010	Aguas Calientes	El Carmen	City	1297	1266	2563
38028003	Aparzo	Humahuaca	Village	28	24	52
38063010	Arrayanal	San Pedro	Village	454	401	855
38063020	Arroyo Colorado	San Pedro	Village	184	194	378
38035010	Bananal	Ledesma	Village	342	347	689
38014020	Barrio El Milagro	El Carmen	Village	658	663	1321
38014030	Barrio La Uniln	El Carmen	Village	362	338	700
38112010	Barrios	Yavi	Village	83	104	187
38035020	Bermejito	Ledesma	Village	83	62	145
38098010	Burcena	Tumbaya	Village	75	64	139
38035030	Caimancito	Ledesma	City	2750	2586	5336
38035040	Calilegua	Ledesma	City	3033	2964	5997
38112020	Cangrejillos	Yavi	Village	87	94	181

GIS AND DATABASES

LINKING SPATIAL AND NON-SPATIAL DATA



Code	Name	Dep	Type	Men	Women	Total
38007020	Abra Pampa	Cochinoca	Capital	4139	4566	8705
38007030	Abralaite	Cochinoca	Village	27	29	56
38007035	Agua de Castilla	Cochinoca	Village	48	48	96
38014010	Aguas Calientes	El Carmen	City	1297	1266	2563
38028003	Aparzo	Humahuaca	Village	28	24	52
38063010	Arrayanal	San Pedro	Village	454	401	855
38063020	Arroyo Colorado	San Pedro	Village	184	194	378
38035010	Bananal	Ledesma	Village	342	347	689
38014020	Barrio El Milagro	El Carmen	Village	658	663	1321
38014030	Barrio La Uniln	El Carmen	Village	362	338	700
38112010	Barrios	Yavi	Village	83	104	187
38035020	Bermejito	Ledesma	Village	83	62	145
38098010	Burcena	Tumbaya	Village	75	64	139
38035030	Caimancito	Ledesma	City	2750	2586	5336
38035040	Callegua	Ledesma	City	3033	2964	5997
38112020	Cangrejillos	Yavi	Village	87	94	181



64.69 W
22.72 S

65.80 W
23.16 S

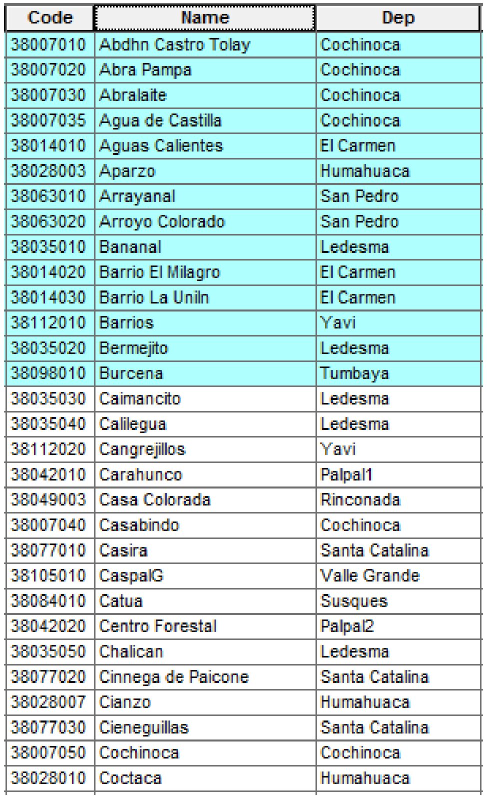
64.59 W
23.73 S

GIS support both!!
This is what makes GIS
a very powerful tool

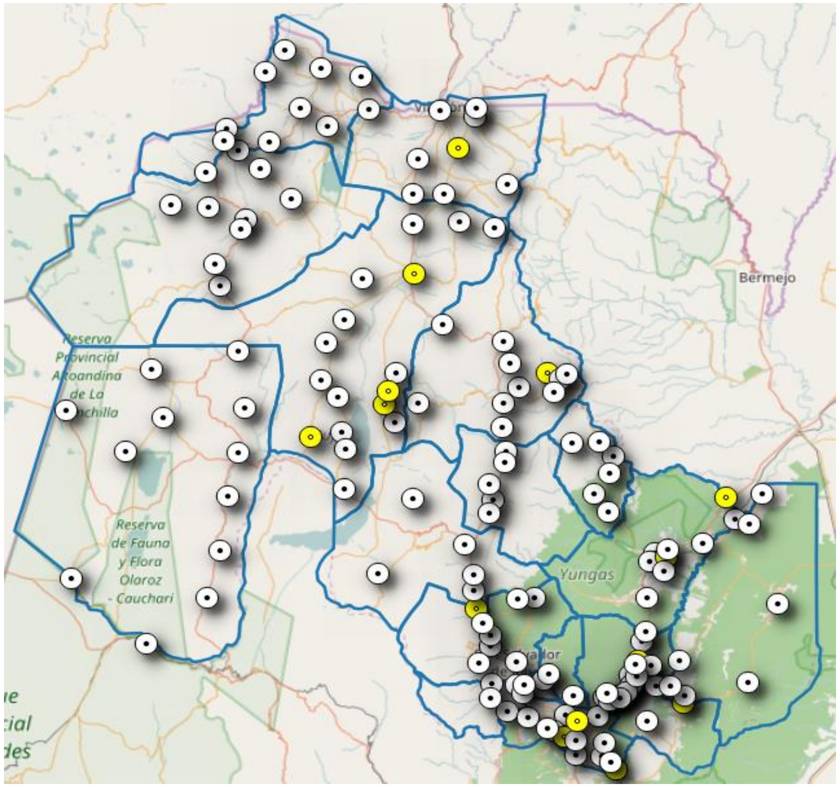
GIS AND DATABASES

LINKING SPATIAL AND NON-SPATIAL DATA

Spatial and non-spatial data in a GIS environment are sometimes stored separately (.shp, .dbf) but dynamically linked



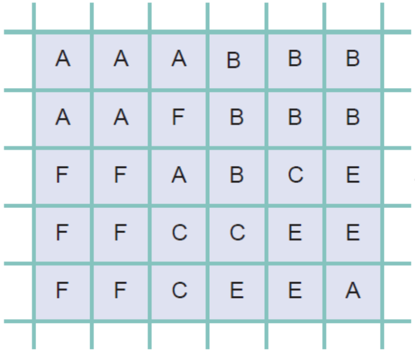
Code	Name	Dep
38007010	Abdhn Castro Tolay	Cochinoca
38007020	Abra Pampa	Cochinoca
38007030	Abralaite	Cochinoca
38007035	Agua de Castilla	Cochinoca
38014010	Aguas Calientes	El Carmen
38028003	Aparzo	Humahuaca
38063010	Arrayanal	San Pedro
38063020	Arroyo Colorado	San Pedro
38035010	Bananal	Ledesma
38014020	Barrio El Milagro	El Carmen
38014030	Barrio La Uniln	El Carmen
38112010	Barrios	Yavi
38035020	Bermejito	Ledesma
38098010	Burcena	Tumbaya
38035030	Caimancito	Ledesma
38035040	Calilegua	Ledesma
38112020	Cangrejillos	Yavi
38042010	Carahunco	Palpal1
38049003	Casa Colorada	Rinconada
38007040	Casabindo	Cochinoca
38077010	Casira	Santa Catalina
38105010	CaspalG	Valle Grande
38084010	Catua	Susques
38042020	Centro Forestal	Palpal2
38035050	Chalican	Ledesma
38077020	Cinnega de Paicone	Santa Catalina
38028007	Cianzo	Humahuaca
38077030	Cieneguillas	Santa Catalina
38007050	Cochinoca	Cochinoca
38028010	Coctaca	Humahuaca


Bermejo
Reserva Provincial Altoandina de La Chinchilla
Reserva de Fauna y Flora Olaroz - Cauchari
Yungas

GIS AND DATABASES

LINKING SPATIAL AND NON-SPATIAL DATA

1 Raster, ***Cell value***
2 Vector, ***Object ID***

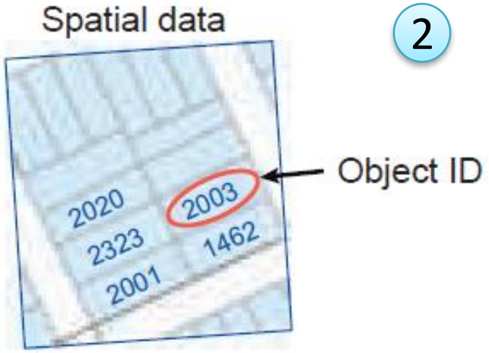


A	A	A	B	B	B
A	A	F	B	B	B
F	F	A	B	C	E
F	F	C	C	E	E
F	F	C	E	E	A

1

LandUseClass	Id	Description	Perc
	A	Primary forest	11.3
	B	Secondary vegetation	25.5
	C	Pasture	31.2
	E	Built-up area	25.5
	F	Rivers, lakes	4.1

2



Parcel	PId	Location	OwnerID
	3421	2001	435
	8871	1462	550
	2109	2323	1040
	1515	2003	245
	3434	2020	486
	6371	1802	950
	2209	3542	1840
	1505	2609	145

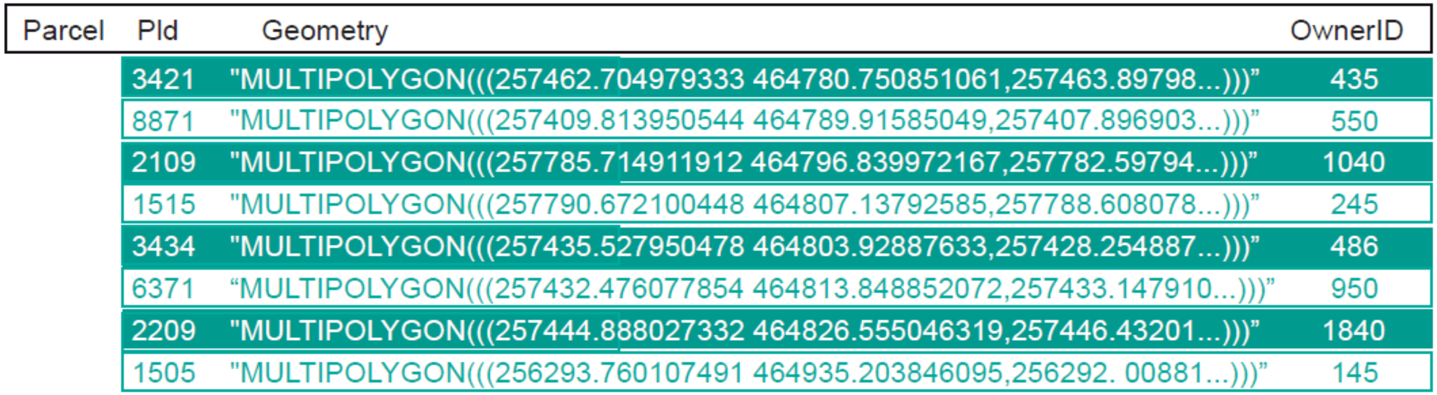
Spatial attribute

UNIVERSITY OF TWENTE.

SPATIAL DATABASES

Databases have evolved over the last 20 years towards supporting more complex data, such as **spatial data**

A **spatial database** allows users to store, query and manipulate collections of **spatial data by means of a new** data type: ***geometry***



Parcel	PId	Geometry	OwnerID
	3421	"MULTIPOLYGON(((257462.704979333 464780.750851061,257463.89798...)))”	435
	8871	"MULTIPOLYGON(((257409.813950544 464789.91585049,257407.896903...)))”	550
	2109	"MULTIPOLYGON(((257785.714911912 464796.839972167,257782.59794...)))”	1040
	1515	"MULTIPOLYGON(((257790.672100448 464807.13792585,257788.608078...)))”	245
	3434	"MULTIPOLYGON(((257435.527950478 464803.92887633,257428.254887...)))”	486
	6371	“MULTIPOLYGON(((257432.476077854 464813.848852072,257433.147910...)))”	950
	2209	"MULTIPOLYGON(((257444.888027332 464826.555046319,257446.43201...)))”	1840
	1505	"MULTIPOLYGON(((256293.760107491 464935.203846095,256292. 00881...)))”	145

SPATIAL DATABASE MANAGEMENT SYSTEM (SDBMS)

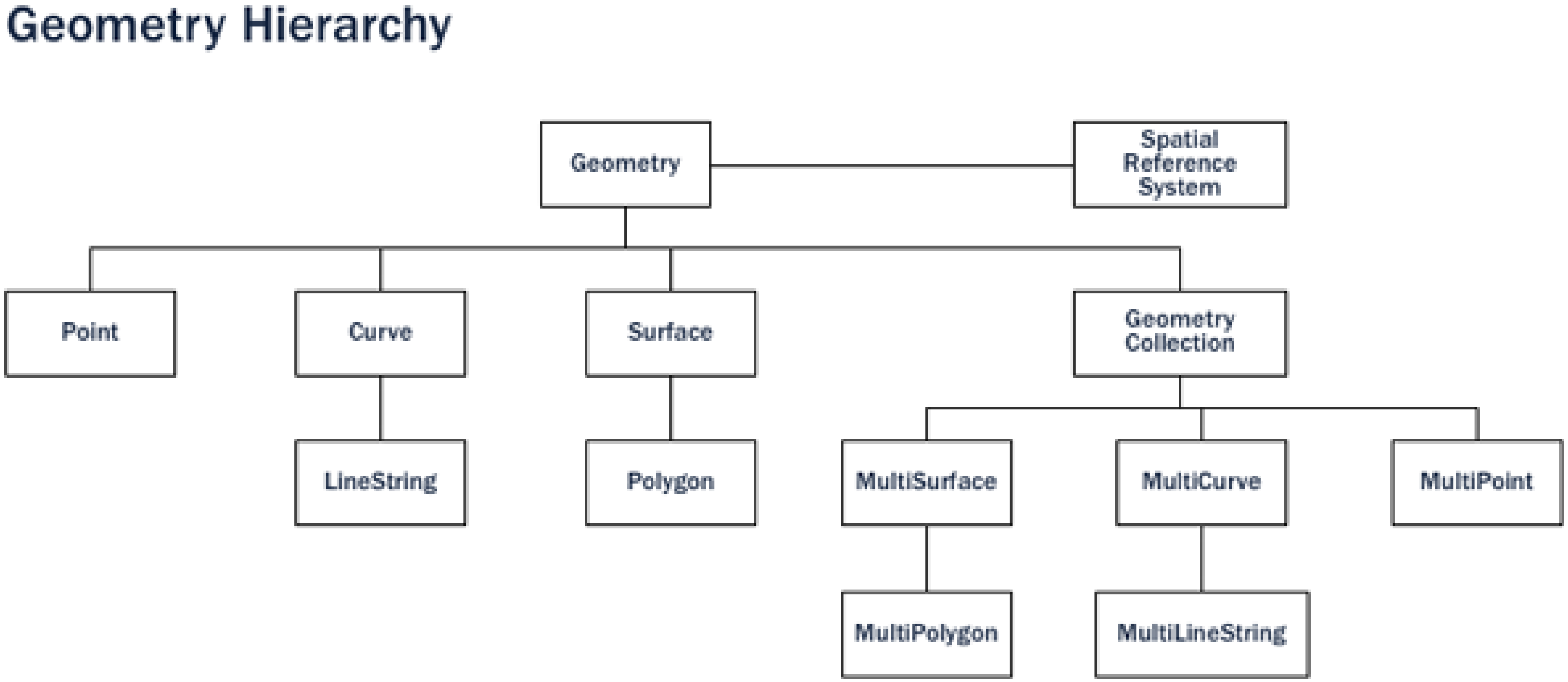
A S(patial)DBMS is a software package that:

- offers **all benefits of a DBMS** for data storage/management
- supports the **storage of spatial data** types – e.g. point, line, or polygon,
- can **manage coordinate systems** and transformations
- **extends querying** and manipulating capabilities of traditional SQL by adding special commands **aimed at spatial data** – e.g.

```
SELECT p.* FROM parcels AS p WHERE
 Area(p.geom)>1000
```

- provide storage of the relationships between features, including the **creation and storage of topological relationships**.

Spatial data types



GIS & SDBMS

A GOOD TEAM

GIS:

- Built-in ‘understanding’ of geographic space,
- Functions for spatial analysis of (almost) any kind, and
- Equipment for efficient map production
- **BUT** lack of fully developed query language to operate on tabular data.

DBMS

- Specifically designed for handling attribute data (i.e. administrative, non-spatial, tabular, thematic)
- Long tradition in multi-user concurrent management of large amounts of data
- **BUT** not support provided for spatial data storage and representation

GIS & SDBMS

A GOOD TEAM

SDBMS:

- Focuses on storage, querying and sharing large **spatial data sets**,
- Understand geometry and topology
- Provide fully-fledged querying language (i.e. SQL)
- Spatial querying functions
- No visualization functionality

PostgreSQL

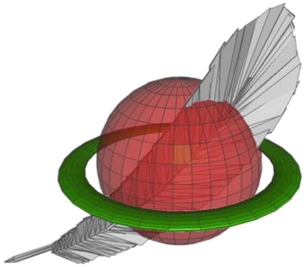
PostGIS

What is PostGIS?

PostGIS turns the PostgreSQL Database Management System into a spatial database by adding support for the three features: spatial types, indexes, and functions.

Alternatives to PostGIS

- ORACLE SPATIAL
- MICROSOFT SQL SERVER
- SPATIALITE
- MYSQL
- ArcSDE



SpatiaLite

ORACLE SPATIAL

Microsoft SQL Server

MySQL Spatial

esri ArcSDE

What applications support PostGIS?

Open/Free	Closed/Proprietary
•Loading/Extracting • Shp2Pgsql • ogr2ogr • Dxf2PostGIS •Web-Based • Mapserver • GeoServer (Java-based WFS / WMS -server) • SharpMap SDK - for ASP.NET 2.0 • MapGuide Open Source (using FDO) •Desktop • uDig • QGIS • mezoGIS • OpenJUMP • OpenEV • SharpMap SDK for Microsoft.NET 2.0 • ZigGIS for ArcGIS/ArcObjects.NET • GvSIG • GRASS	•Loading/Extracting • Safe FME Desktop Translator/Converter •Web-Based • Ionic Red Spider (now ERDAS) • Cadcorp GeognoSIS • Iwan Mapserver • MapDotNet Server • MapGuide Enterprise (using FDO) • ESRI ArcGIS Server 9.3+ •Desktop • Cadcorp SIS • Microimages TNTmips GIS • ESRI ArcGIS 9.3+ • Manifold • GeoConcept • MapInfo (v10) • AutoCAD Map 3D (using FDO)

Installation

Use Application Stack Builder

Direct download from http://postgis.net/install

- You must install appropriate version

Creating a Spatial Database

Create a database in pgAdmin

- Name it sdb

Select the new database

Open Query Tool

Load the PostGIS spatial extension in the database by:

```
CREATE EXTENSION postgis;
```

Confirm that PostGIS is installed:

```
SELECT postgis_full_version();
```

Points - example

```
CREATE TABLE city (
  city_id serial PRIMARY KEY,
   name varchar(100),
  geom geometry(POINT)
);


INSERT INTO city (name, geom)
VALUES (
   ‘Emschede’,
  ST_GeomFromText('POINT(1 -1)')) ;
```

Loading sample data

Use PostGIS Shape and DBF Loader to add shapefiles to the database

Data (New York):

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

Data

nyc_census_blocks
  - blkid, popn_total, boroname, geom
- nyc_streets
  - name, type, geom
nyc_subway_stations
  - name, geom
- nyc_neighborhoods
  - name, boroname, geom

Simple SQL

Retrieve data

```
SELECT name FROM nyc_neighborhoods;

SELECT * FROM nyc_neighborhoods;

SELECT geom FROM  nyc_census_blocks;

SELECT ST_AsText(geom) FROM  nyc_census_blocks;

SELECT ST_Area(geom) FROM  nyc_census_blocks;
```

Function list

Within the database, geometries are stored on disk in a format only used by the PostGIS program.

PostGIS supports emitting and consuming geometries in a large number of formats:

- 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

Question:

- Describe each of above formats in three sentences.

Function list

sum(expression) aggregate to return a sum for a set of records

count(expression) aggregate to return the size of a set of records

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

ST_X(point) returns the X ordinate

ST_Y(point) returns the Y ordinate

ST_Length(linestring) 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

ST_Area(geometry) returns the area of the polygons

ST_NRings(geometry) returns the number of rings (usually 1, more if there are holes)

ST_ExteriorRing(polygon) returns the outer ring as a linestring

Function list

ST_InteriorRingN(polygon, integer) returns a specified interior ring as a linestring
ST_Perimeter(geometry) returns the length of all the rings
ST_NumGeometries(multi/geomcollection) returns the number of parts in the collection
ST_GeometryN(geometry, integer) returns the specified part of the collection
ST_GeomFromText(text) returns geometry
ST_AsText(geometry) returns WKT text
ST_AsEWKT(geometry) returns EWKT text
ST_GeomFromWKB(bytea) returns geometry
ST_AsBinary(geometry) returns WKB bytea
ST_AsEWKB(geometry) returns EWKB bytea
ST_GeomFromGML(text) returns geometry
ST_AsGML(geometry) returns GML text
ST_GeomFromKML(text) returns geometry
ST_AsKML(geometry) returns KML text
ST_AsGeoJSON(geometry) returns JSON text
ST_AsSVG(geometry) returns SVG text

Functions for intersect, disjoint, overlap, cross, and touch

ST_Intersects(geometry A, geometry B) returns TRUE if the two shapes have any space in common, i.e., if their boundaries or interiors intersect.
The opposite of ST_Intersects is **ST_Disjoint(geometry A , geometry B)**. If two geometries are disjoint, they do not intersect, and vice-versa.
For multipoint/polygon, multipoint/linestring, linestring/linestring, linestring/polygon, and linestring/multipolygon comparisons, **ST_Crosses(geometry A, geometry B)** returns 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.
ST_Touches(geometry A, geometry B) returns TRUE if either of the geometries' boundaries intersect or if only one of the geometry's interiors intersects the other's boundary.

Example

Let’s take our Broad Street subway station and determine its neighborhood using the **ST_Intersects** function:

```
SELECT name, ST_AsText(geom)
FROM nyc_subway_stations
WHERE name = 'Broad St';
--result: POINT(583571 4506714)
SELECT name, boroname
FROM nyc_neighborhoods
WHERE ST_Intersects(geom,
ST_GeomFromText('POINT(583571 4506714)',26918));
```

Example

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
      );
```

Example

```
SELECT
  subways.name AS subway_name,
  neighborhoods.name AS neighborhood_name,
  neighborhoods.boroname AS borough

FROM
       nyc_neighborhoods AS neighborhoods, nyc_subway_stations AS
subways

WHERE subways.name = 'Broad St’ AND ST_Contains(neighborhoods.geom,
subways.geom);
```

Other sample questions

Questions on nyc_census_blocks:

What is the population of the City of New York?

What is the population of the Bronx? (boroname is ‘The Bronx’)

What is the total length of streets (in kilometers) in New York City?

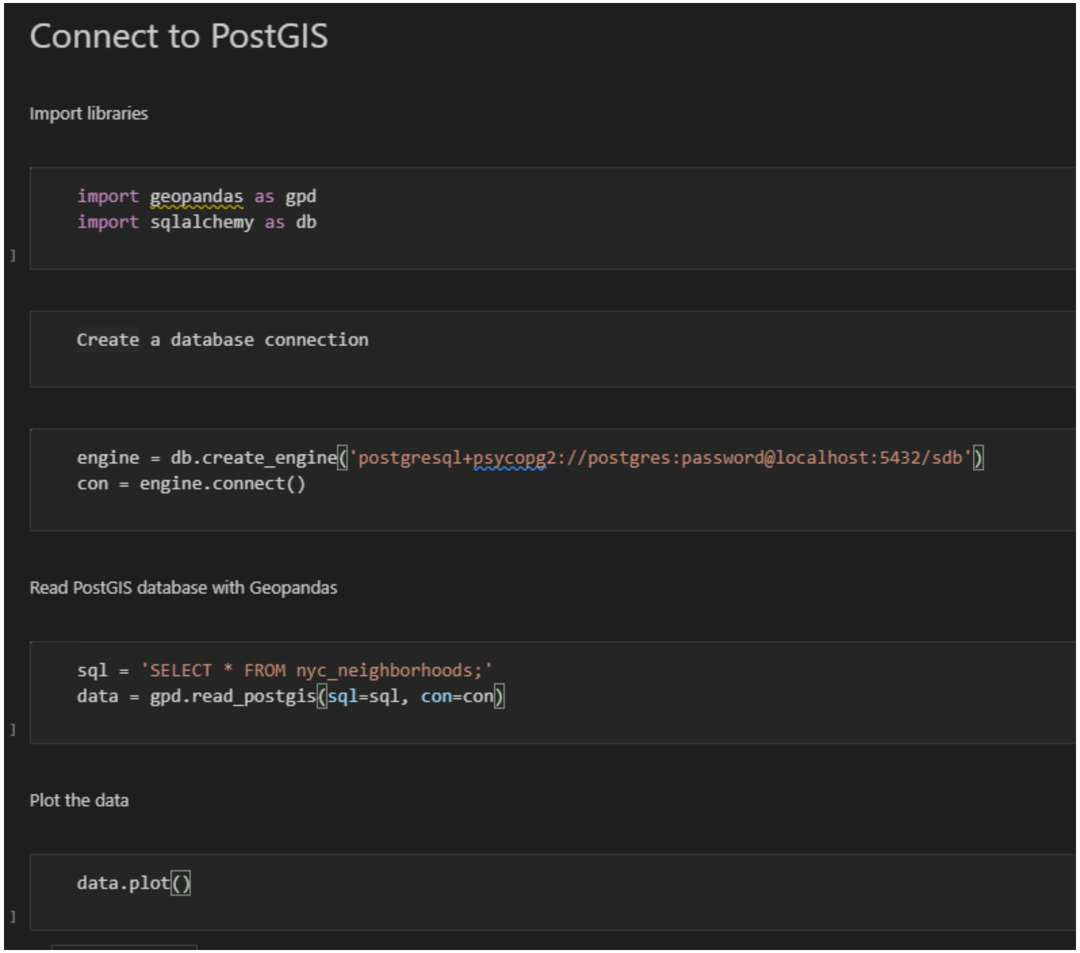
PostGIS and GIS Software

+

PostGIS and Python

Install sqlalchemy, geopandas

conda install geopandas



Connect to PostGIS

Import libraries

```
import geopandas as gpd
import sqlalchemy as db
```

```
Create a database connection
```

```
engine = db.create_engine('postgresql+psycopg2://postgres:password@localhost:5432/sdb')
con = engine.connect()
```

Read PostGIS database with Geopandas

```
sql = 'SELECT * FROM nyc_neighborhoods;'
data = gpd.read_postgis(sql=sql, con=con)
```

Plot the data

```
data.plot()
```

QUESTIONS?