

Stefanakis, E., 2014. *Geographic Databases and Information Systems*. CreateSpace Independent Publ. [In English], pp.386.

Get a copy from [Amazon](#)

Chapter 15

Geographic Database Systems

Emmanuel Stefanakis

<http://www2.unb.ca/~estef/>

Geographic Data in GIS

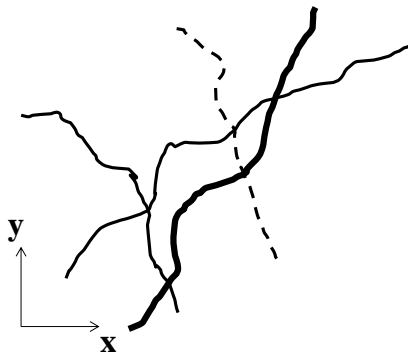
- Two main components...
 - Spatial (geometry)
 - Non-Spatial (attribute data)



Streets : Table			
	Name	Type	Street_ID
▶	Street A	RD	1
	Street B	RD	2
	Street C	RD	3
	Service Road	RD	5
	Street A2	AVE	16
*			

Geographic Data in GIS

- How are those stored in a GIS ?...
 - A common approach is to use two systems (hybrid architecture)



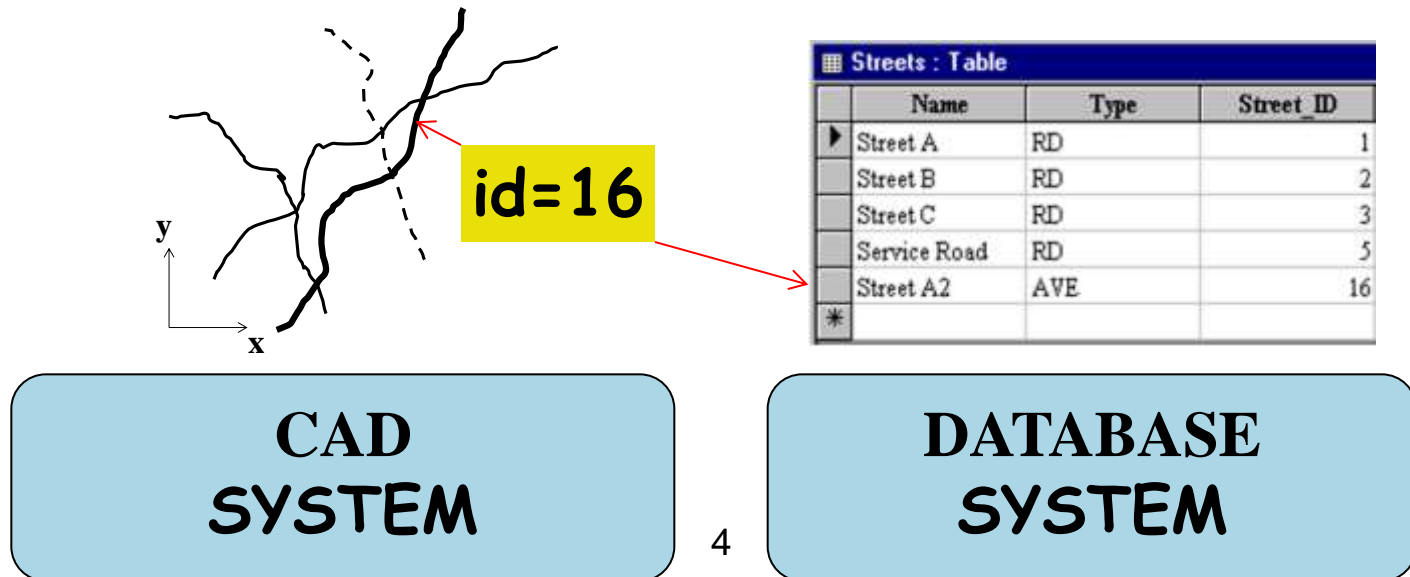
**CAD
SYSTEM**

Streets : Table			
	Name	Type	Street_ID
▶	Street A	RD	1
	Street B	RD	2
	Street C	RD	3
	Service Road	RD	5
	Street A2	AVE	16
*			

**DATABASE
SYSTEM**

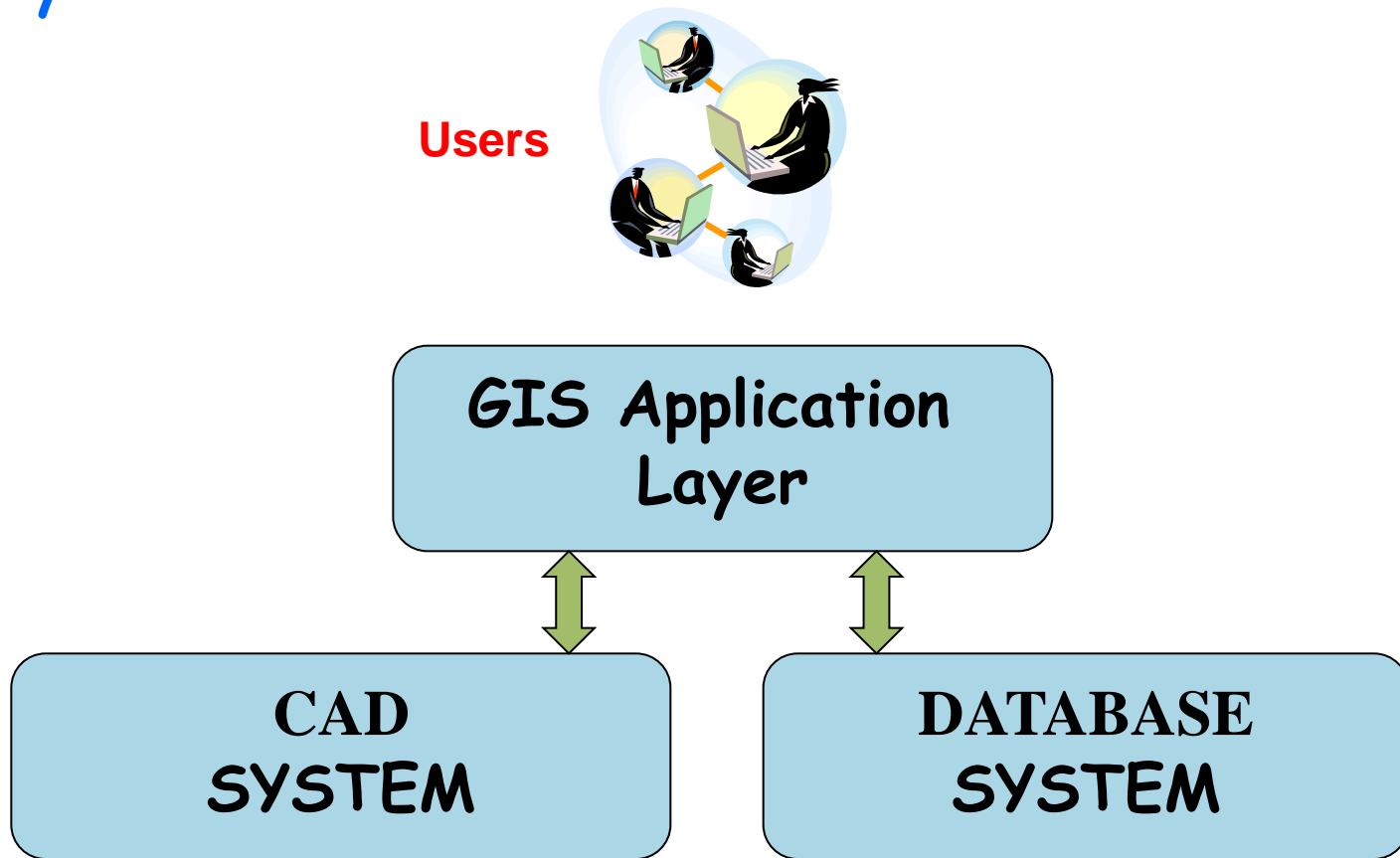
Geographic Data in GIS

- Data related to the same entity...
 - is split and stored into the two systems
 - Re-structuring → through common identifiers



Geographic Data in GIS

- Hybrid Architecture...



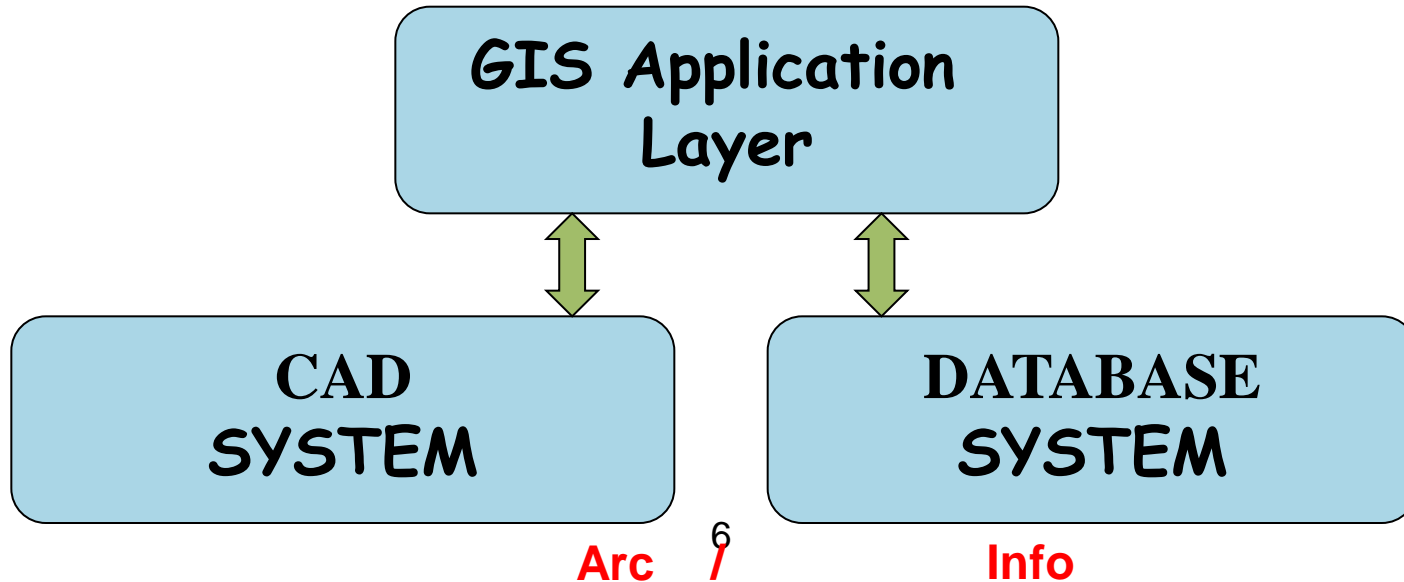
Geographic Data in GIS

- Hybrid Architecture...

Users



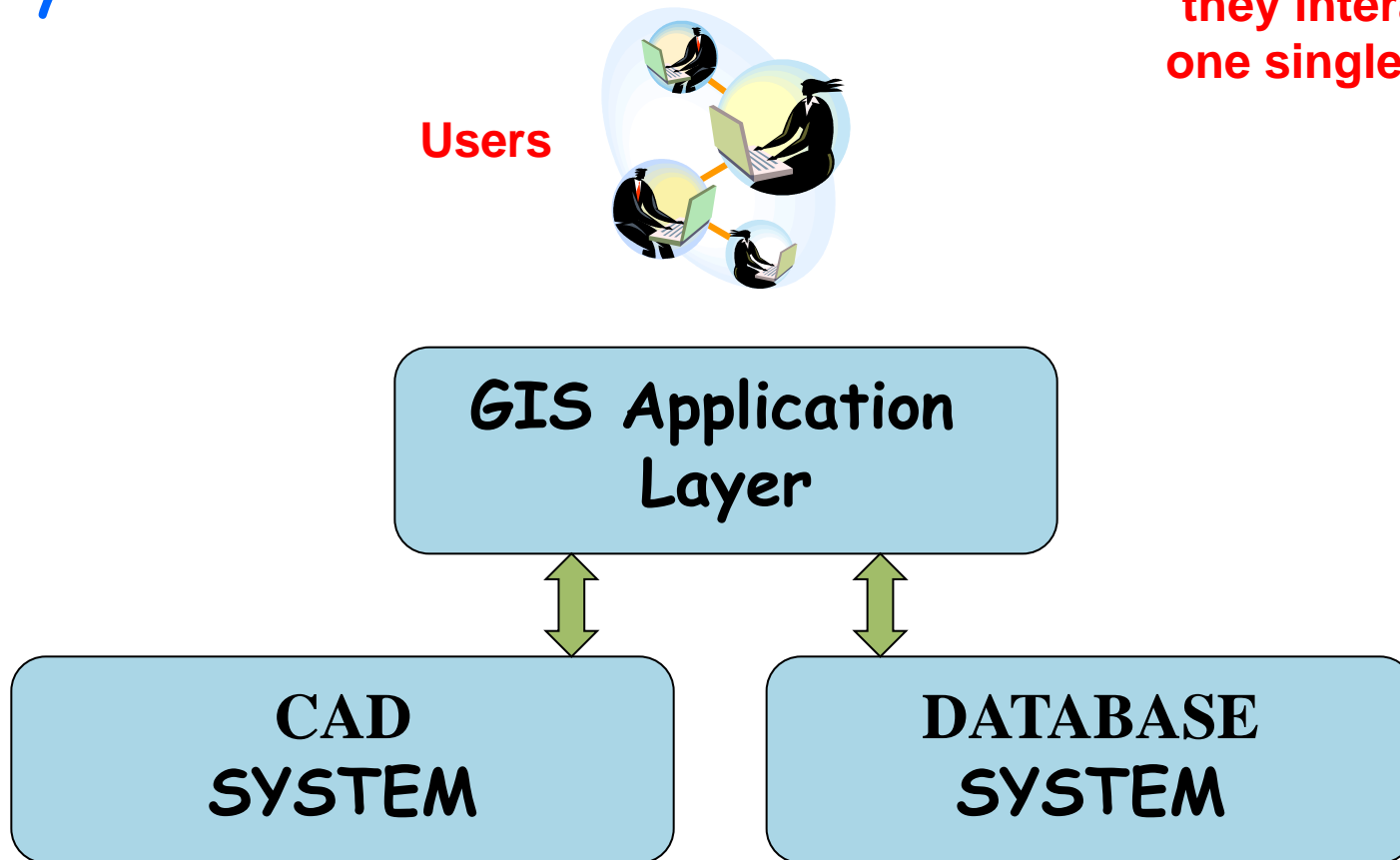
The application layer is responsible of making the two sub-systems work together



Geographic Data in GIS

- Hybrid Architecture...

The users feel that they interact with one single system.

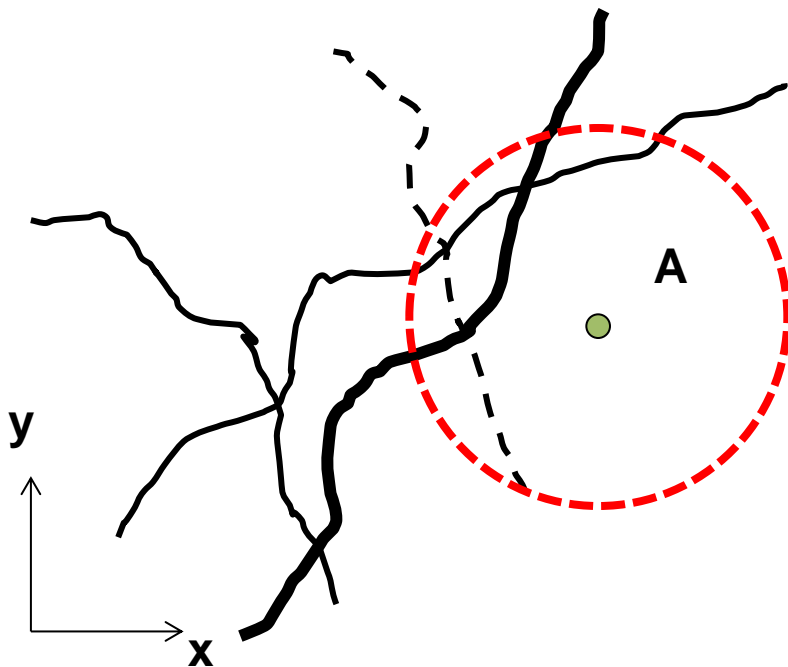


Querying Geographic Data

- Three types of queries...
 - Spatial
 - based on geometries
 - Non-spatial
 - based on attributes
 - Combined
 - both spatial and non-spatial predicates

Where is the processing performed...

- Spatial Query: Find all streets that are less than 5 km away from point A.

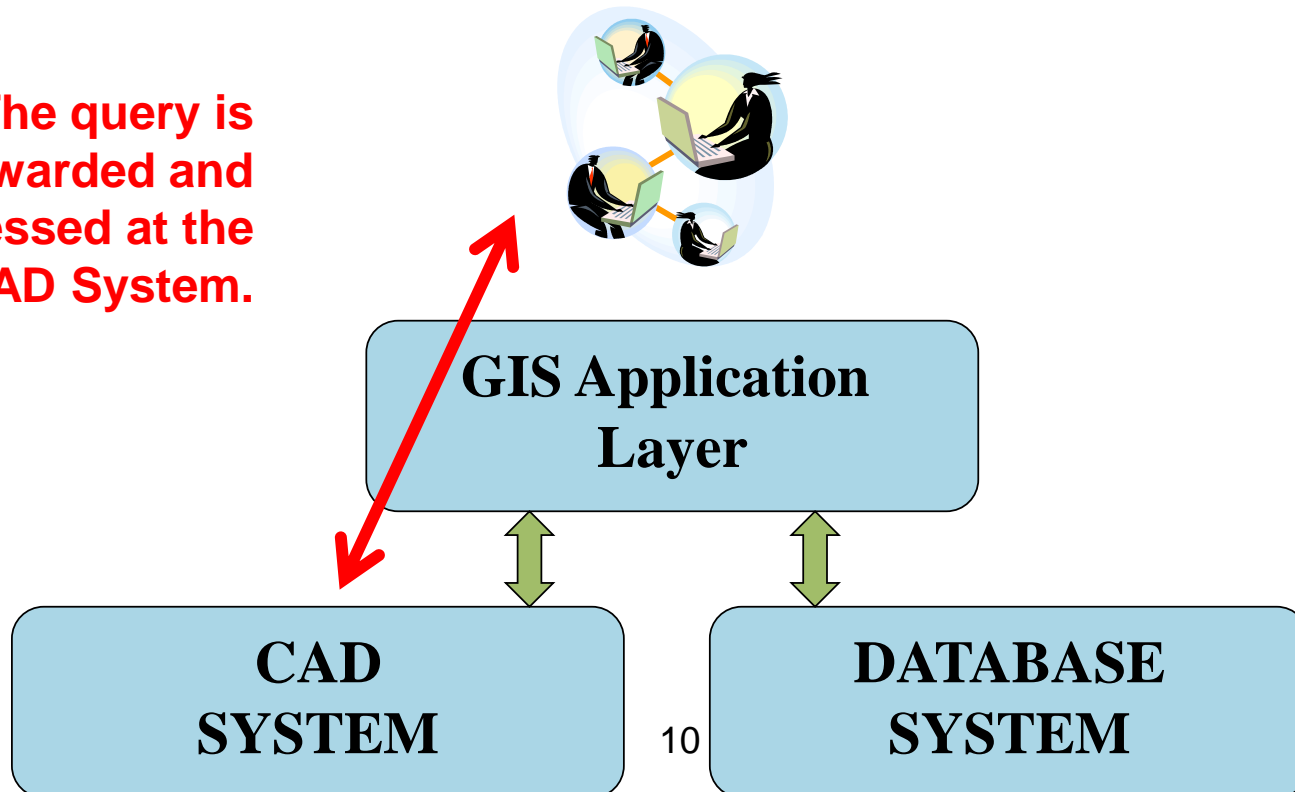


Streets : Table			
	Name	Type	Street_ID
▶	Street A	RD	1
	Street B	RD	2
	Street C	RD	3
	Service Road	RD	5
	Street A2	AVE	16
*			

Where is the processing performed...

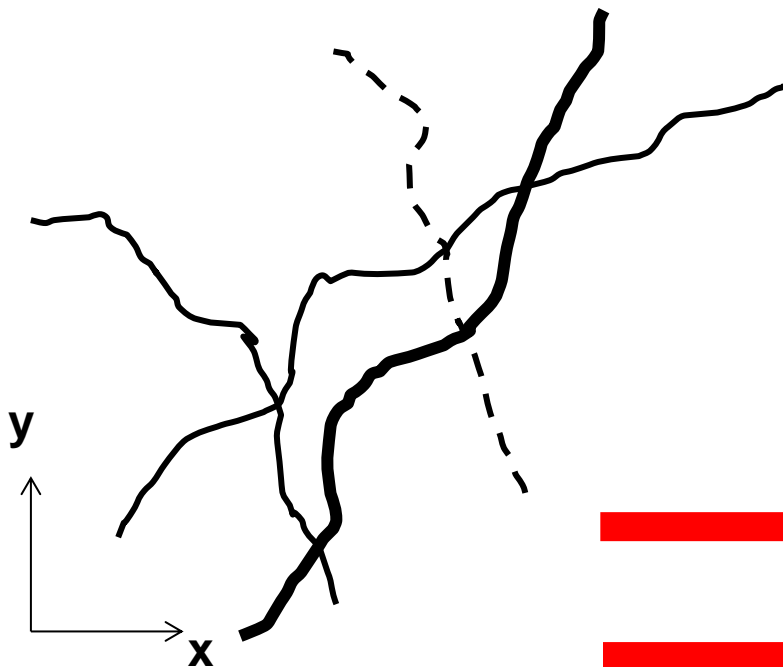
- Spatial Query: Find all streets that are less than 5 km away from point A.

The query is forwarded and processed at the CAD System.

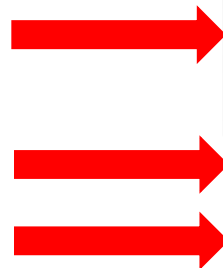


Where is the processing performed...

- Non-Spatial Query: Find all streets that are of type AVE (avenues).

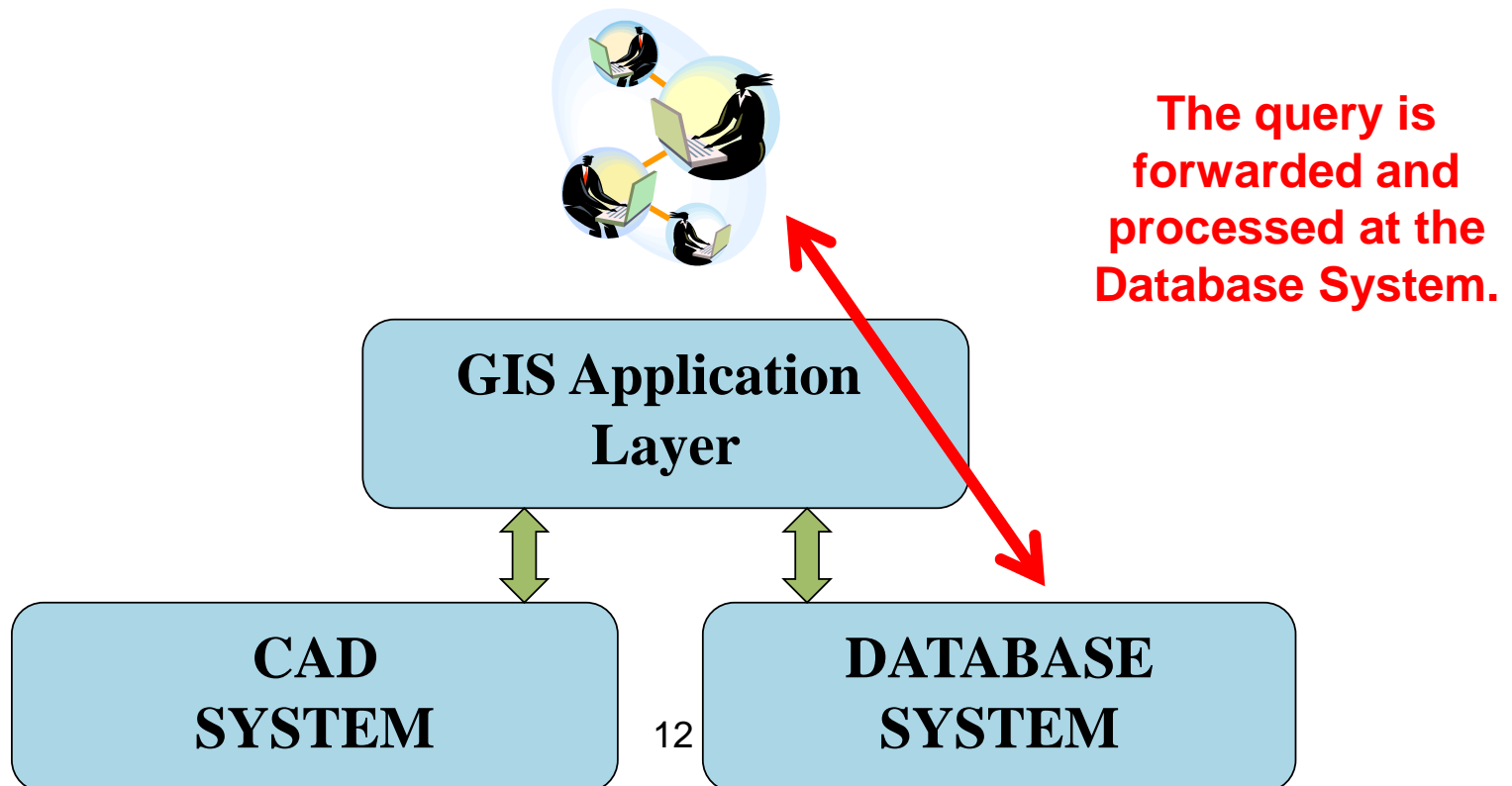


Streets : Table			
	Name	Type	Street_ID
▶	Street A	RD	1
	Street B	RD	2
	Street C	RD	3
	Service Road	RD	5
	Street A2	AVE	16
*			



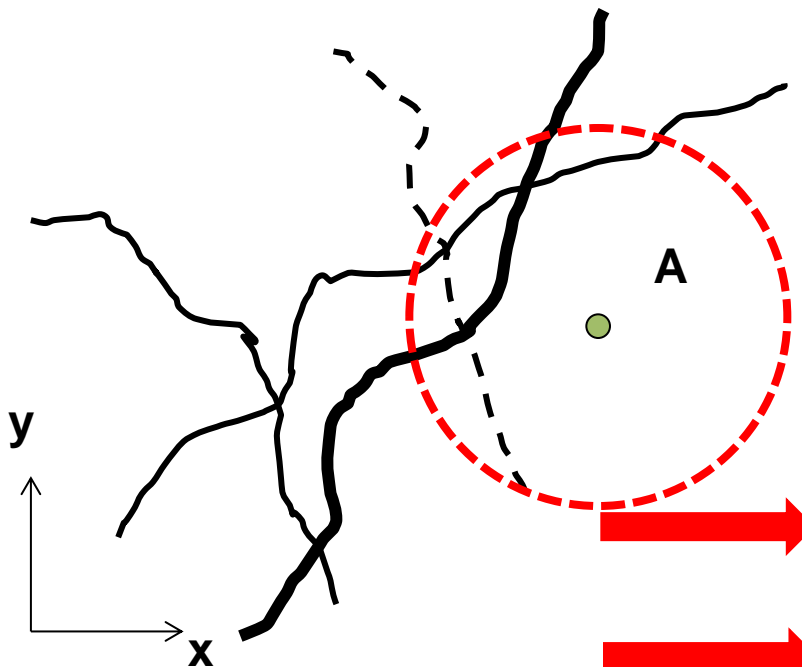
Where is the processing performed...

- Non-Spatial Query: Find all streets that are of type AVE (avenues).



Where is the processing performed...

- Combined Query: Find all avenues that are less than 5 km away from point A.

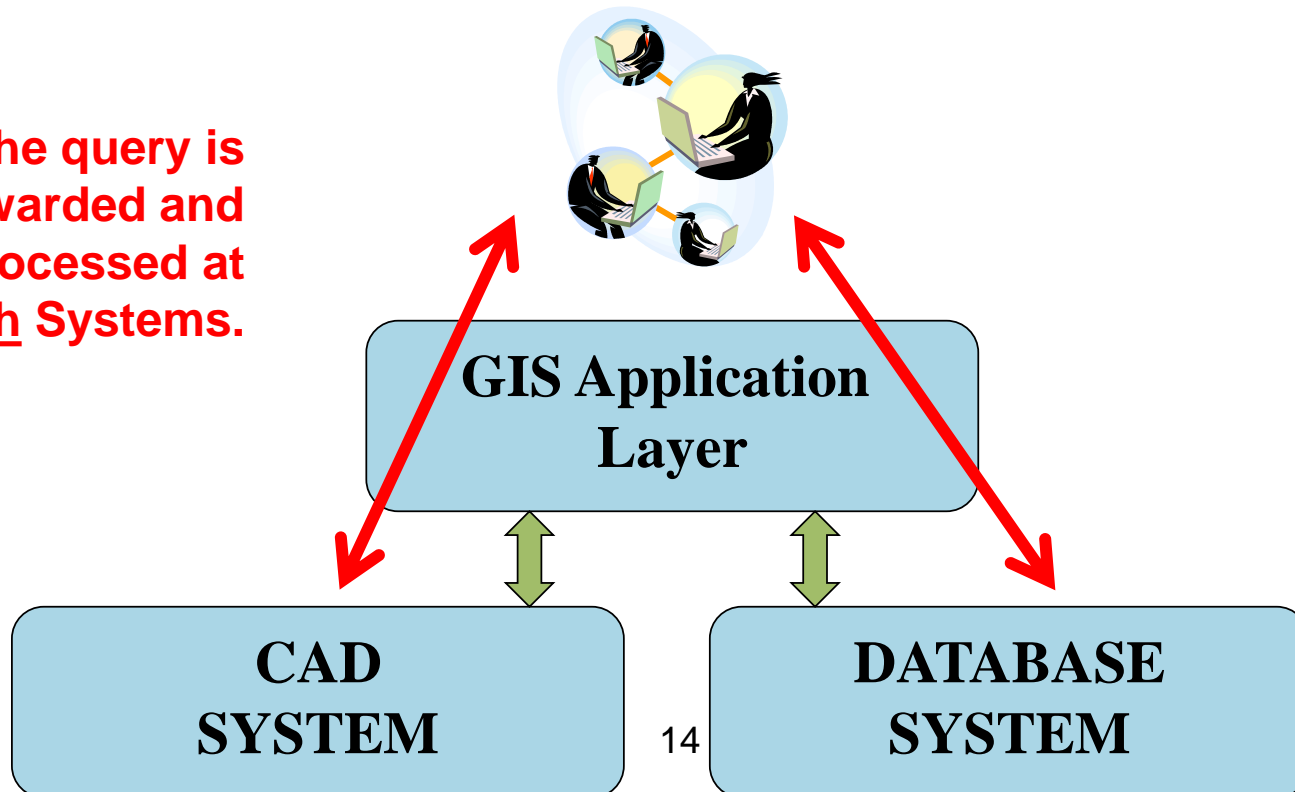


Streets : Table			
	Name	Type	Street_ID
▶	Street A	RD	1
	Street B	RD	2
	Street C	RD	3
	Service Road	RD	5
	Street A2	AVE	16
*			

Where is the processing performed...

- Combined Query: Find all avenues that are less than 5 km away from point A.

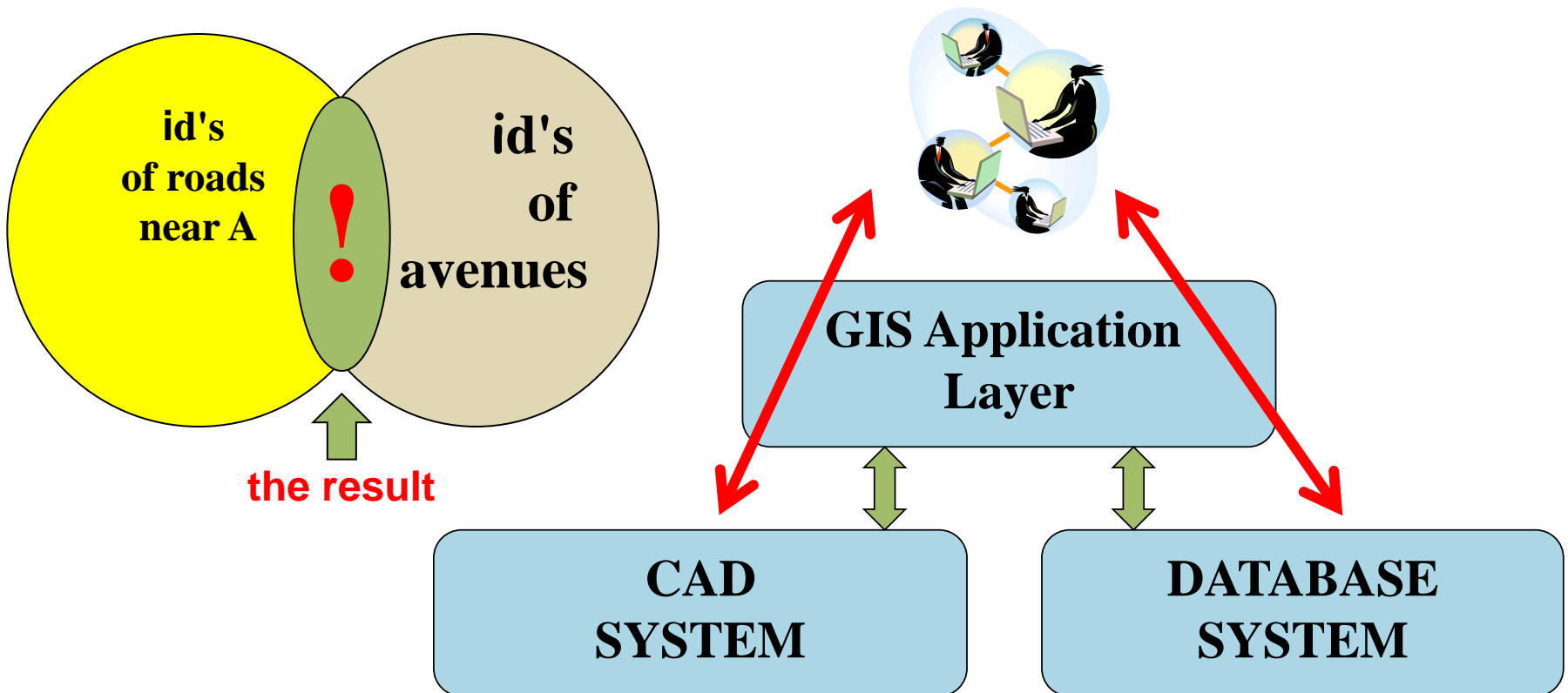
The query is forwarded and processed at both Systems.



Where is the processing performed...

- Combined Query...

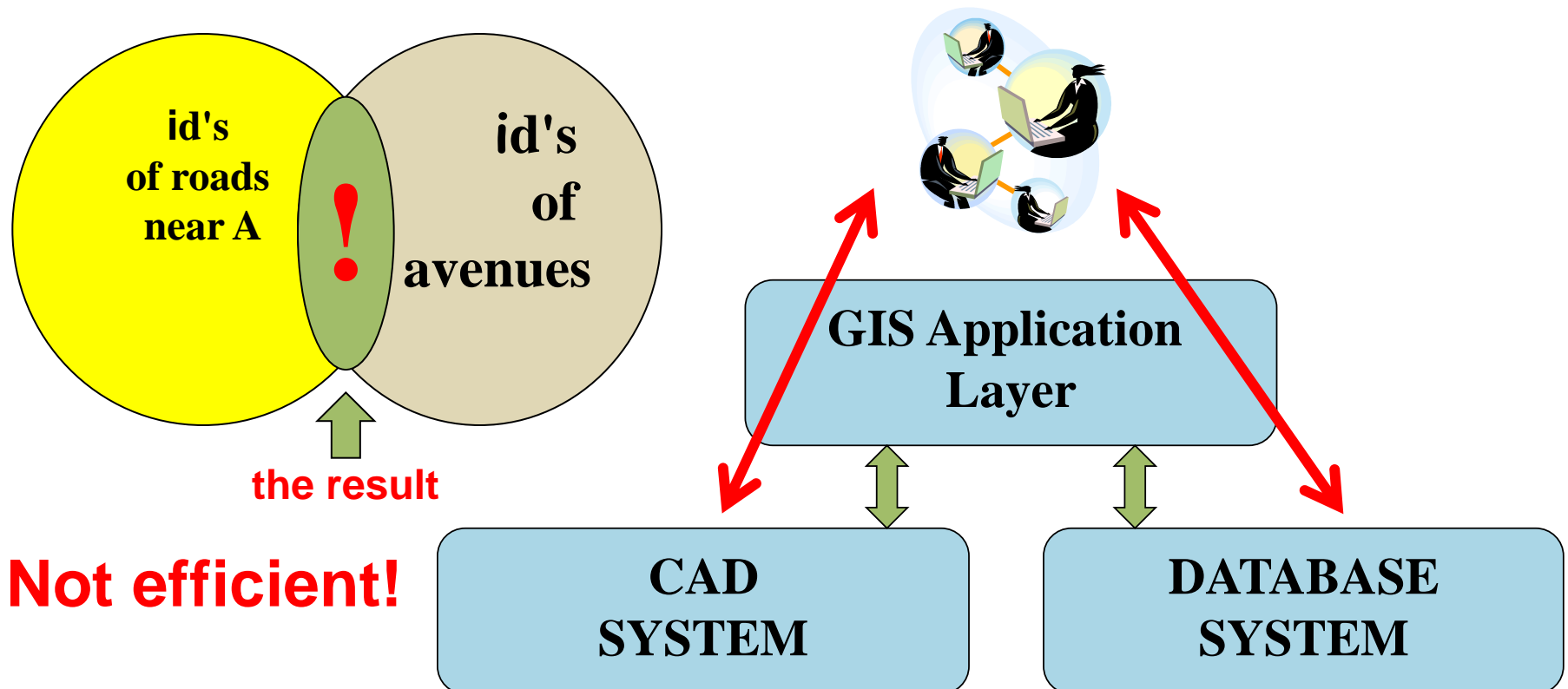
- Appl. Layer is responsible to find the intersection.



Where is the processing performed...

- Combined Query...

- Appl. Layer is responsible to find the intersection.



Geographic Data in GIS

- A similar Architecture (Geodatabase)...



Users

No CAD System; single DB.

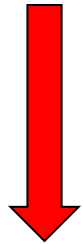
Geometries are stored as binary fields in the traditional database.

GIS Application Layer



DATABASE SYSTEM

Relational technology



Streets : Table				
	Name	Type	Street_ID	Geometry1
▶	Street A	RD	1	Long binary data
	Street B	RD	2	Long binary data
	Street C	RD	3	Long binary data
	Service Road	RD	5	Long binary data
	Street A2	AVE	16	Long binary data
*				

Geographic Data in GIS

- A similar Architecture (Geodatabase)...



Users

The spatial processing is performed at the application layer.

Not effective either !

GIS Application Layer



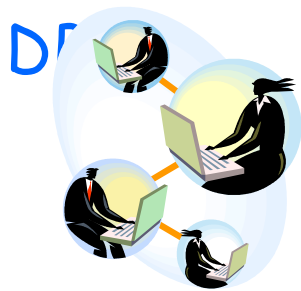
DATABASE SYSTEM

Relational technology

Streets : Table				
	Name	Type	Street_ID	Geometry1
▶	Street A	RD	1	Long binary data
	Street B	RD	2	Long binary data
	Street C	RD	3	Long binary data
	Service Road	RD	5	Long binary data
	Street A2	AVE	16	Long binary data
*				

Geographic Data in GIS

- A more efficient Architecture (Spatial



Users

All processing is performed in the database (at the lowest level)

Very effective !

GIS Application Layer



SPATIAL DATABASE SYSTEM

id	type	the_geom
1	EU_CODE	LINestring(593743.88 3916620,593715.56 3916622.25,59369
2	EU_CODE	LINestring(614009.56 3909609,613916.88 3909639,613797.7
3	EU_CODE	LINestring(631576.56 3905657.5,631569.06 3905659,631556
4	EU_CODE	LINestring(632869.63 3905207.75,632824.81 3905188.75,63
5	EU_CODE	LINestring(627792.160346413 3906020.79383269,627790.80
6	EU_CODE	LINestring(616933.661270548 3909441.95504963,616925.13
7	EU_CODE	LINestring(616048.586419265 3909484.07791837,616003.28
8	EU_CODE	LINestring(594452.885137049 3912461.15567697,594423.22

Object relational technology

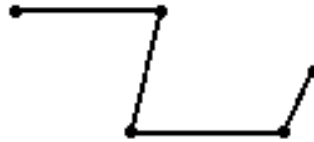
Define the Spatial Schema

- New Geometry types...

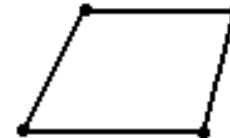
Point



Line String



Polygon



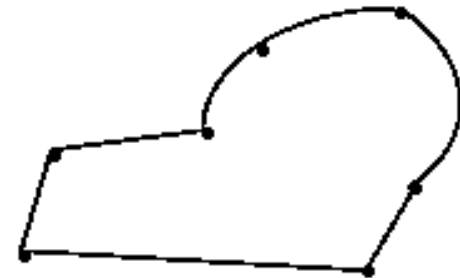
Arc Line String



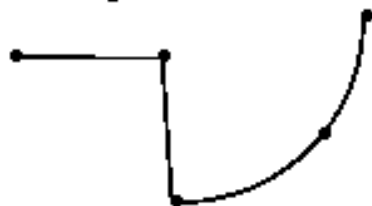
Arc Polygon



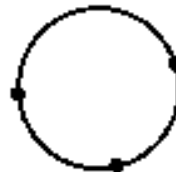
Compound Polygon



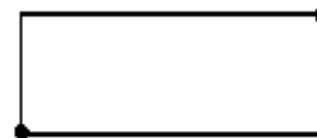
Compound Line String



Circle



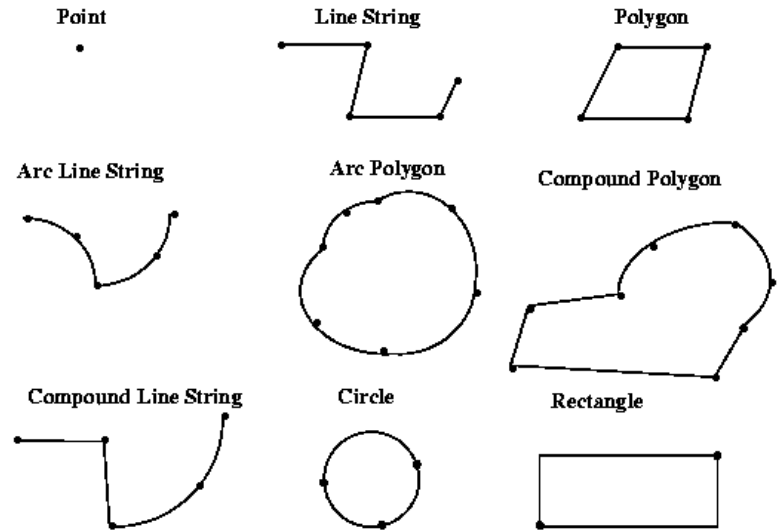
Rectangle



Define the Spatial Schema

- New Functions...

- Distance()
- Area()
- Intersection()
- ...



- New Index Structures (Spatial indices)

- R-tree
- Quadtree

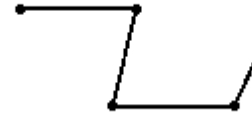
Define the Spatial Schema

- New Geometry types / Functions...

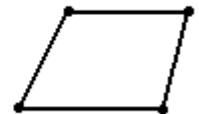
Point



Line String



Polygon



```
CREATE TYPE POINT AS OBJECT (  
    X NUMBER,  
    Y NUMBER,  
    Z NUMBER,  
    FUNCTIONS (  
        DISTANCE (U: POINT, V:POINT)  
        RETURNS NUMBER  
    )  
);
```

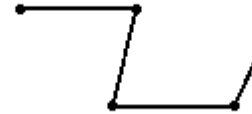
Define the Spatial Schema

- New Geometry types / Functions...

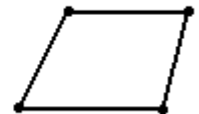
Point



Line String



Polygon



```
CREATE TABLE TOWNS (  
    ID          NUMBER,  
    NAME        VARCHAR(20),  
    LOCATION    POINT,  
);
```

```
SELECT  DISTANCE(T1,T2)  
FROM    TOWNS  
WHERE   T1.NAME = 'Fredericton'  
AND    T2.NAME = 'Oromocto'
```

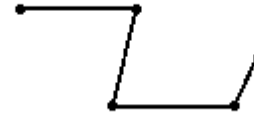
Define the Spatial Schema

- New Geometry types / Functions...

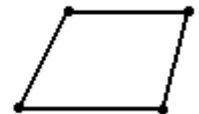
Point



Line String



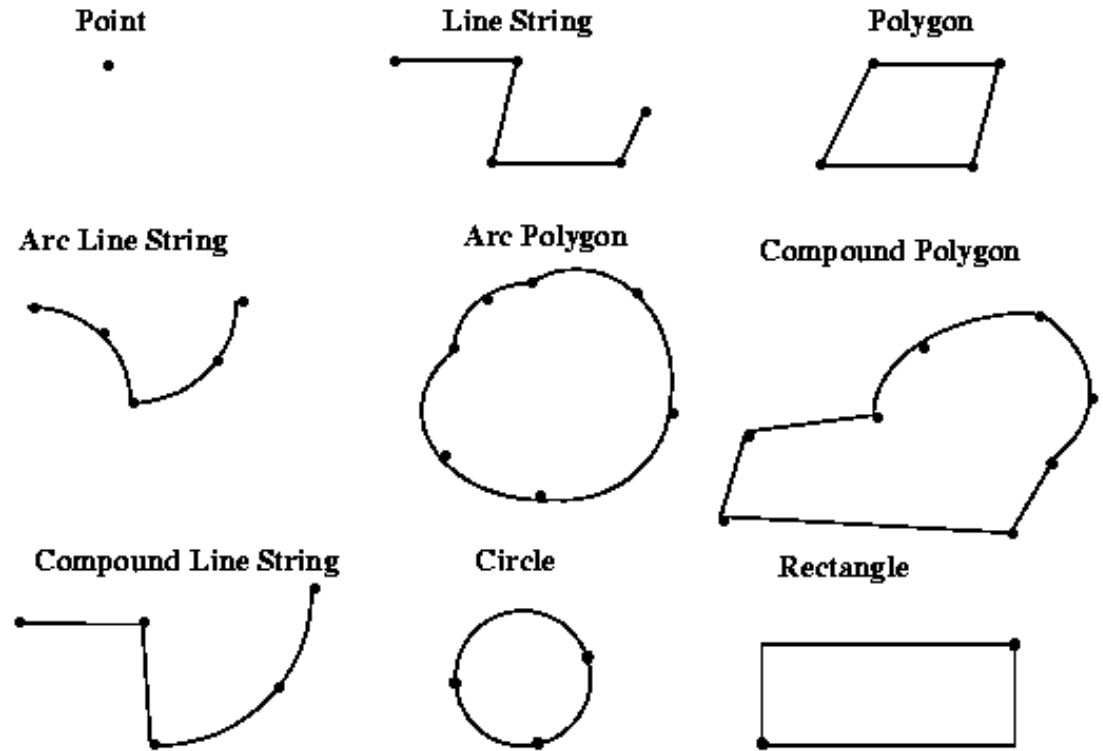
Polygon



```
CREATE TYPE LINESTRING AS OBJECT (  
  P: ARRAY OF POINT,  
  FUNCTIONS (  
    LENGTH (L: LINESTRING)  
    RETURNS NUMBER  
  )  
);
```


Define the Spatial Schema

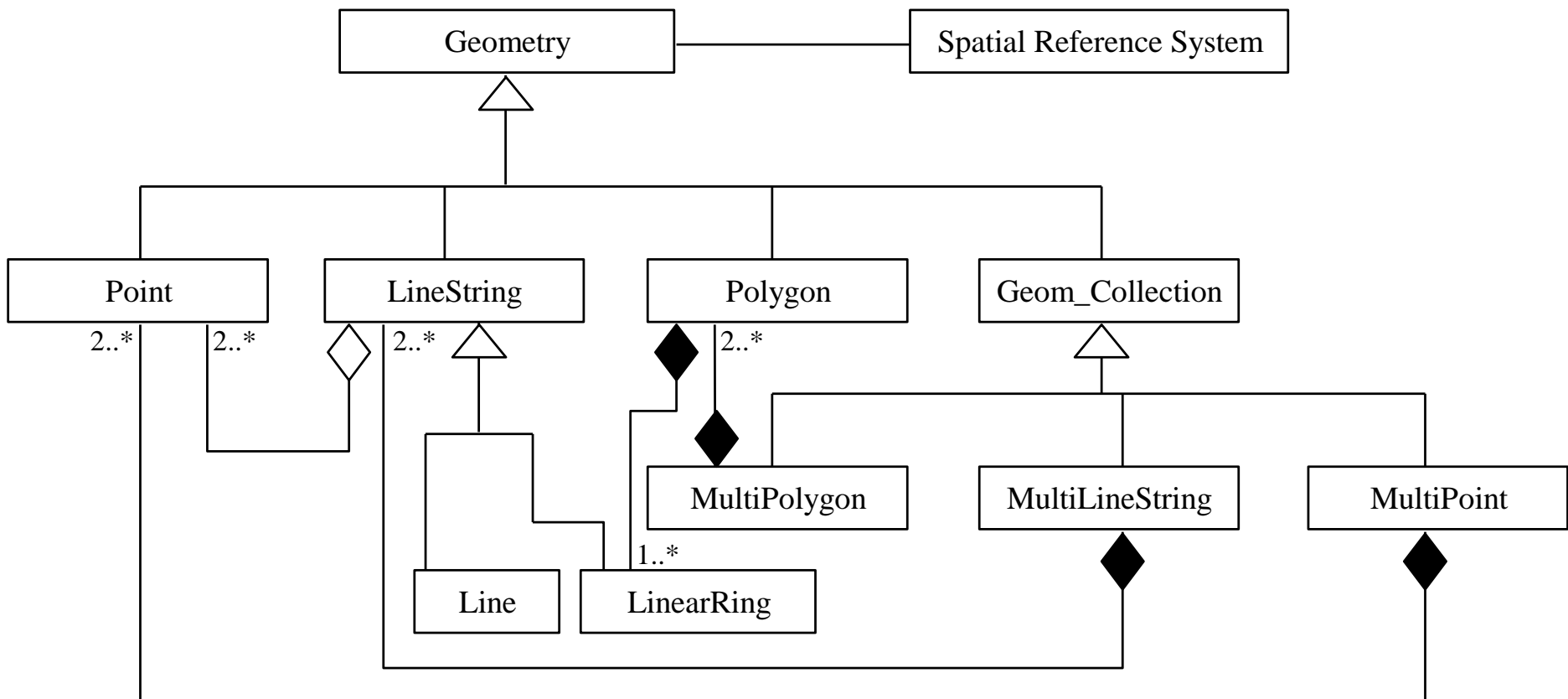
- New Geometry types / Functions...



Too much work !!!

Define the Spatial Schema

- The work has been done for us !



Define the Spatial Schema

- The work has been done for us !



O-Rel DBMS

PostGIS



SPATIAL MODEL

**Application model
(built on top of that)**

Define the Spatial Schema

- The work has been done for us !



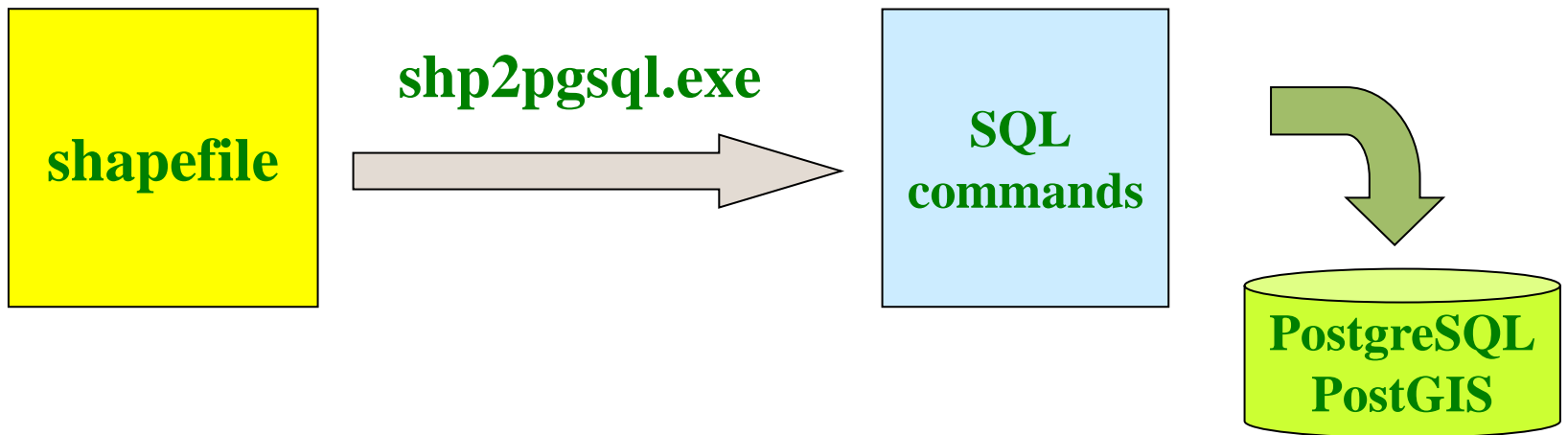
O-Rel DBMS

ORACLE SPATIAL
AND GRAPH
SPATIAL MODEL

Application model
(built on top of that)

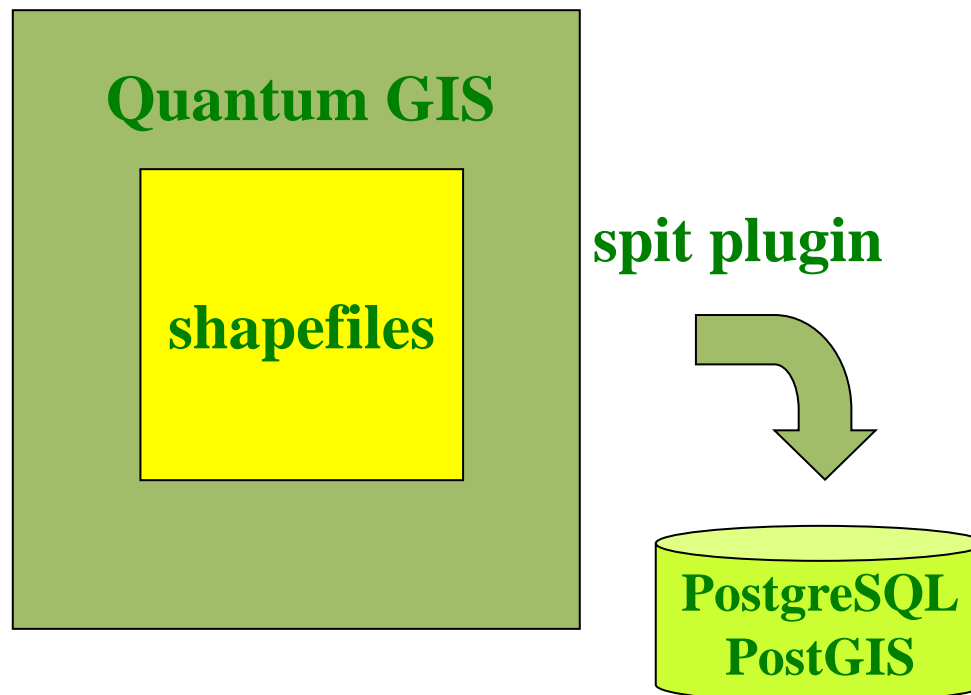
Create a GeoDB in PostgreSQL/PostGIS

- Use the **shp2pgsql** utility offered by PostGIS (it is located inside bin directory)
 - It converts a shapefile to an sqlscript (set of SQL commands)

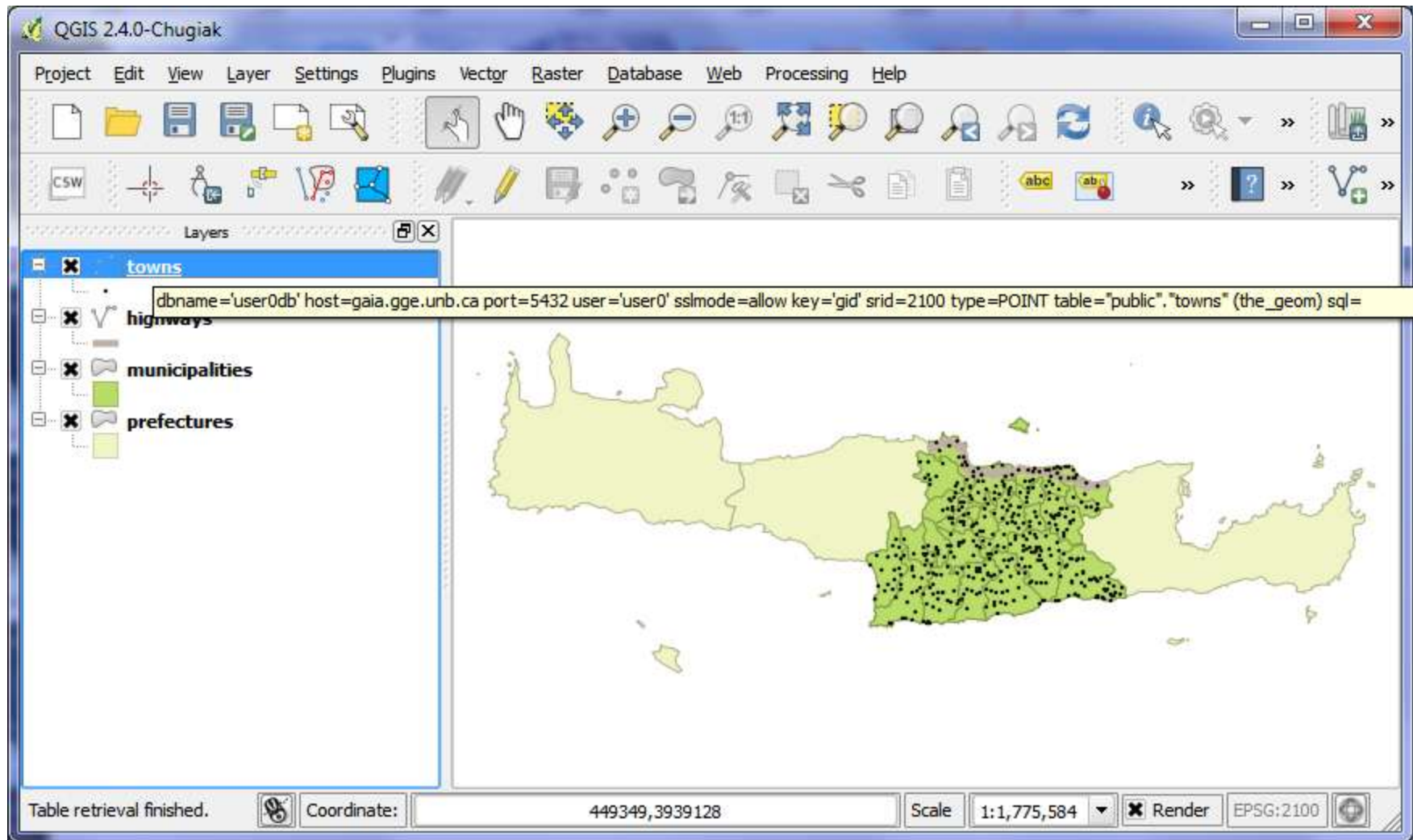


Create a GeoDB in PostgreSQL/PostGIS

- **spit** plugin (Quantum GIS)
 - Import Shapefiles to PostgreSQL (spit)



Connect and Visualize the PostGIS Layers from Quantum



1. Select Population from Prefectures
2. Sum-up Population from Municipalities

The screenshot shows the Quantum GIS 1.3.0-Mimas interface. The map displays the island of Crete with municipalities highlighted in green and prefectures in yellow. Two attribute tables are open, one for prefectures and one for municipalities.

Attribute table - prefectures :: Features total: 4, filtered: 4, s...

gid	ID	NAME	POP_01
0	1	Irakleio	292489
1	2	Lassithi	76319
2	3	Rethymno	81936
3	4	Chania	150387

Attribute table - municipalities :: Features total: 26, filtered: ...

gid	ID	NAME	POP_01
0	1	841 Irakleio	137711
1	2	842 Agia Varvara	5310
2	8	848 Gorgolaini	3171
3	11	851 Episkopi	2533
4	3	843 Arkalochori	10897
5	4	844 Archanes	4548
6	5	845 Asterousia	6303
7	6	846 Viannos	6463

municipalities **prefectures**

1. Select Population from Prefectures
 2. Sum-up Population from Municipalities
-

SELECT "NAME", "POP_01" FROM prefectures;

**SELECT sum("POP_01") FROM municipalities;
(Prefecture: Irakleio)**



The screenshot shows the phpPgAdmin interface. The query results table is as follows:

NAME	POP_01
Irakleio	292489
Lassithi	76319
Rethymno	81936
Chania	150387

4 row(s)
Total runtime: 0.000 ms
SQL executed.



The screenshot shows the phpPgAdmin interface. The query results table is as follows:

sum
292489

1 row(s)
Total runtime: 0.000 ms
SQL executed.

Edit SQL | Download

Towns in the municipality of Mallia.

```
SELECT r."NAME" as municipality, m. "NAME" as town
FROM municipalities AS r, towns AS m
WHERE r.the_geom && m.the_geom
AND ST_contains(r.the_geom, m.the_geom)
AND r."NAME" = 'Mallia';
```



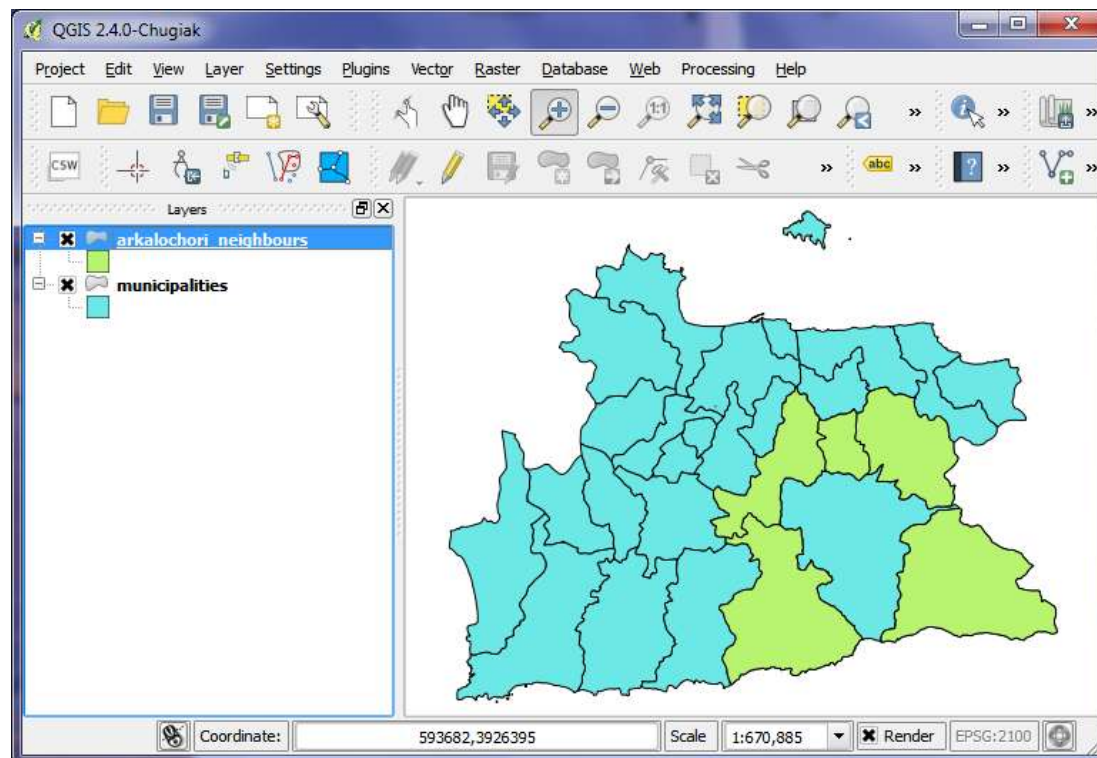
The screenshot shows the phpPgAdmin interface in a web browser. The browser address bar displays the URL: `gaia.gge.unb.ca:8088/phpPgAdmin/sql.php`. The interface indicates that PostgreSQL 9.3.5 is running on `127.0.0.1:5432` and the user is logged in as `user0`. The query results are displayed in a table with the following data:

municipality	town
Mallia	Stalida
Mallia	Malia
Mallia	Mochos
Mallia	Krasio

Below the table, it indicates "4 row(s)" and "Total runtime: 0.000 ms". A "back to top" link is visible in the bottom right corner.

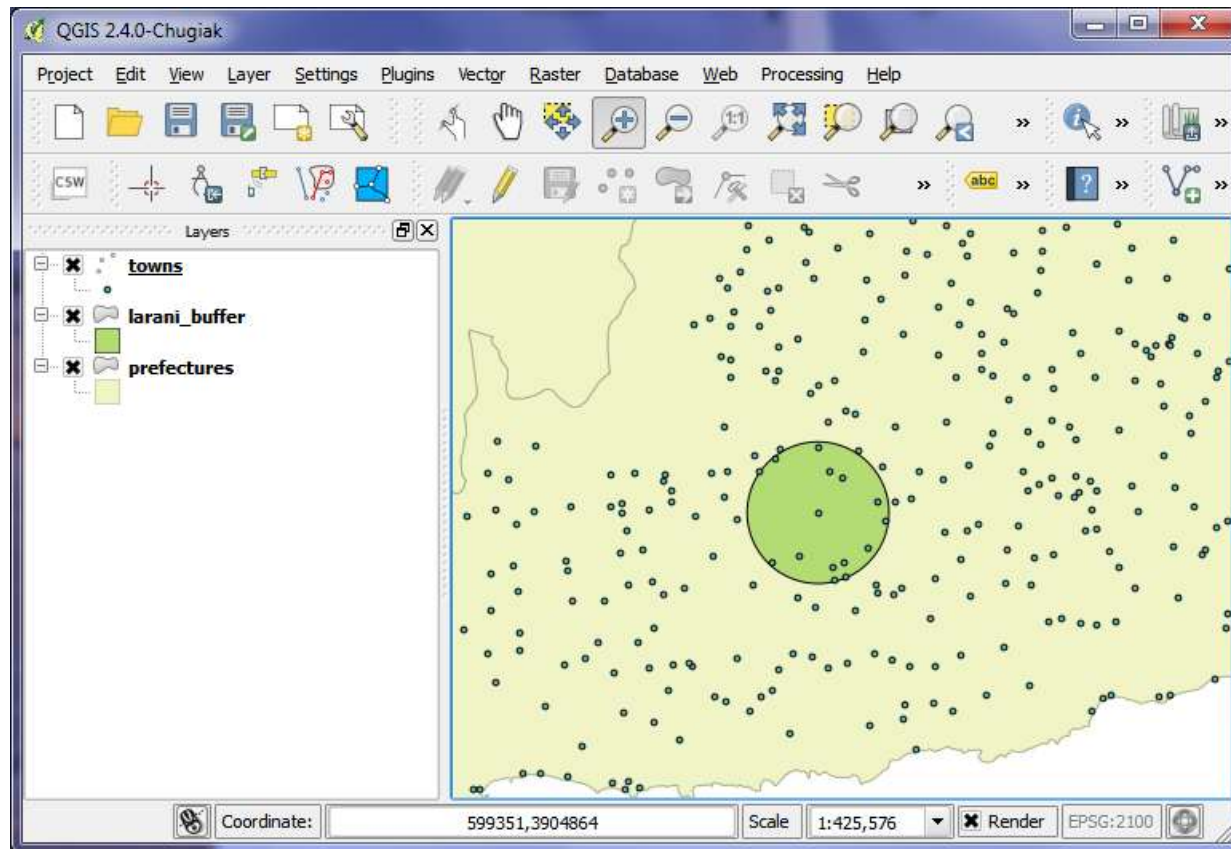
Municipalities sharing a boundary with the municipality of Arkalochori (report the geometry)

```
SELECT n.gid, n.the_geom INTO Arkalochori_neighbours  
FROM municipalities as m, municipalities as n  
WHERE m."NAME" = 'Arkalochori'  
AND ST_Touches(m.the_geom, n.the_geom);
```



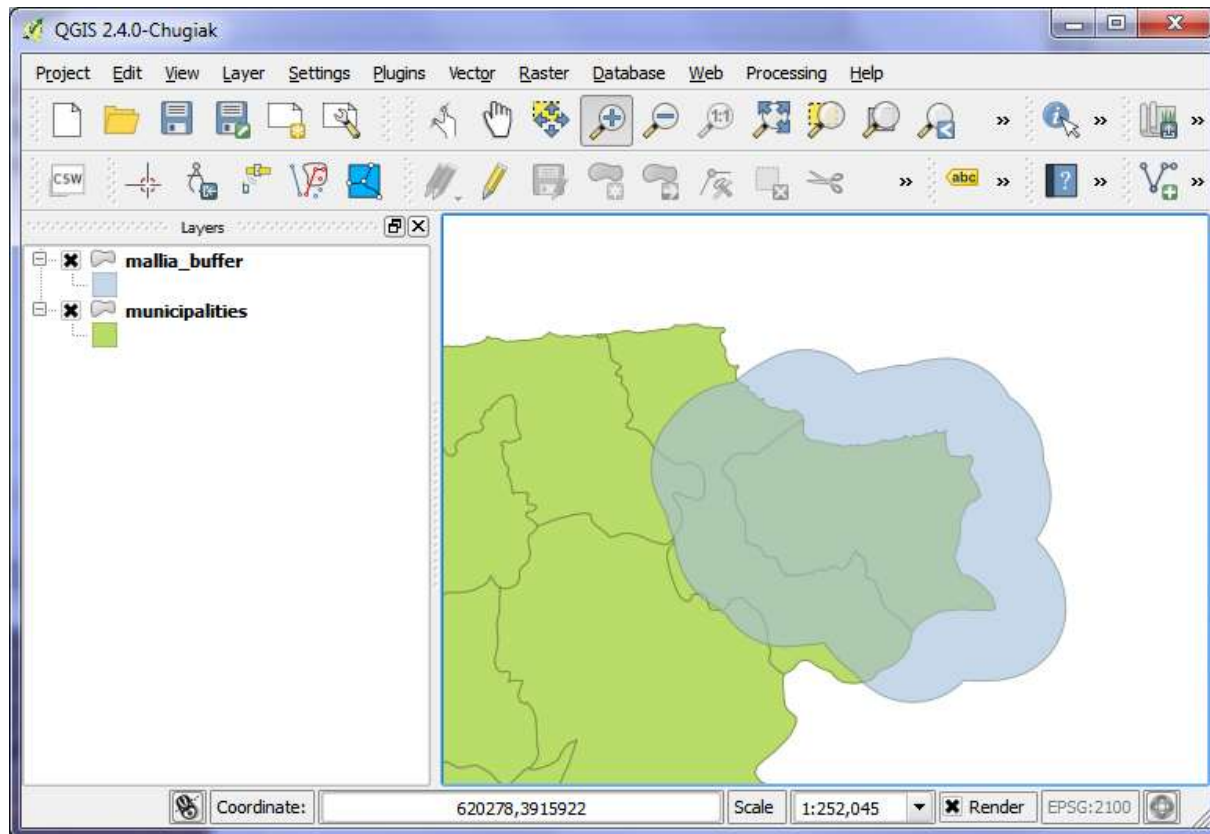
Buffer Zone of 5km around Larani

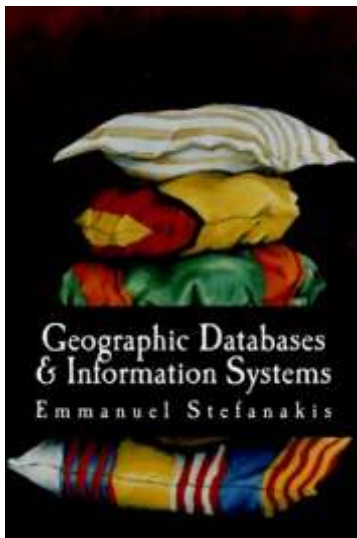
```
SELECT "ID", ST_Buffer(the_geom, 5000, 16) INTO Larani_Buffer  
FROM towns  
WHERE "NAME" = 'Larani';
```



Buffer Zone of 3km around Mallia

```
SELECT ST_Buffer(the_geom, 3000, 16) INTO mallia_buffer  
FROM municipalities  
WHERE "NAME" = 'Mallia';
```





Stefanakis, E., 2014. *Geographic Databases and Information Systems*. CreateSpace Independent Publ. [In English], pp.386.

Get a copy from [Amazon](#)

Chapter 15

Geographic Database Systems

Emmanuel Stefanakis

<http://www2.unb.ca/~estef/>