



**HAL**  
open science

## OpenJUMP workshop: GIS client and Spatial Data Infrastructure Lausanne

Erwan Bocher, Arnaud Lepetit, Uggo Taddei

► **To cite this version:**

Erwan Bocher, Arnaud Lepetit, Uggo Taddei. OpenJUMP workshop: GIS client and Spatial Data Infrastructure Lausanne. 2006. hal-01887531

**HAL Id: hal-01887531**

**<https://hal.science/hal-01887531>**

Submitted on 4 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# OpenJUMP workshop



## GIS client and Spatial Data Infrastructure

Lausanne

12/09/2006

# Table of contents

---

<b>Introduction</b>	page 3
<b>Using OpenJUMP</b>	page 6
1. Loading and displaying data	page 6
2. Working with alphanumeric data	page 10
3. Thematic maps	page 15
4. Exporting view	page 17
5. Querying data	page 18
<b>Using OpenJUMP in SDI context</b>	page 22
1. OpenJUMP and PostGIS	page 22
1.1. Create a database and load PostGIS extension	page 22
1.2. Using PostGIS	page 24
2. OpenJUMP advancing to a OGC client	page 39

# Introduction

## Summary

The aim of the workshop is to show how OpenJUMP can be used to help building and managing a Spatial Data Infrastructure (SDI). The figure 1 illustrate the logical system of our SDI and figure 2 the technical application.

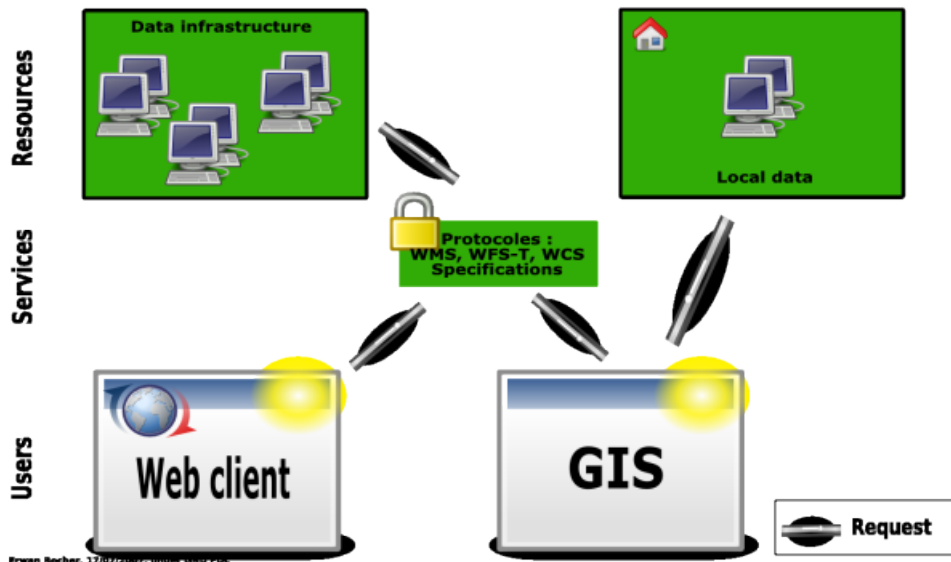


Figure 1

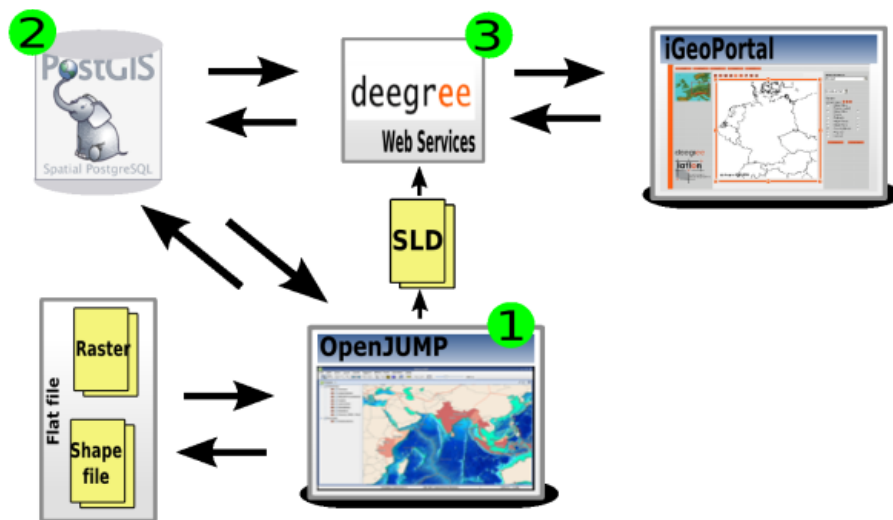


Figure 2

Here, you will learn how to create a PostGIS database and how to add data to OpenJUMP. Visualization, styling and geoprocessing with OpenJUMP will be illustrated with simple examples.

This document is organized in two steps which allow a progressive learning of OpenJUMP.

### 1. Using OpenJUMP

This part guides you through some basic OpenJUMP skills. You'll learn how to :

- load and display data,
- working with alphanumeric data,
- thematic maps,
- exporting views
- querying and geoprocessing

### 2. Using OpenJUMP in SDI context

This part is the most important step in the workshop. We will demonstrate how it's possible to create and manage a simple SDI with OpenJUMP. Two sections are developed.

First in the section "OpenJUMP and Postgis" we use PostgreSQL and PostGIS spatial functions to increase spatial analysis. You can create advanced spatial processing and querying single or multiple data source with SQL command line. You'll learn how to :

- searching for strings, values,
- building expressions with more than one criteria,
- searching and creating new data by their location.

Secondly, the section "OpenJUMP advancing to a OGC client" is intended as an introduction to web services. Connection with deegree WFS protocol will be shown and a simple client browser will be used.

## Materials

We need to have :

- OpenJUMP foss4g edition available at <http://www.projet-sigle.org>,
- PostgreSQL database and postGIS extension available at <http://www.postgresql.org/>,
- Deegree tools (web services and web browser) available at <http://www.deegree.org>,
- OpenOffice 2.0 available at <http://www.openoffice.org/>,
- Inkscape available at <http://www.inkscape.org/>.

And spatial data, available at <http://www.projet-sigle.org>

# Data

Zip file	Data	Geometry	Description
frenchprovinces	frenchprovinces	polygon	Regional country
bzhdepartements	bzhdepartements	polygon	Sub french country after province
municipalities22	municipalities22	polygon	Municipalities for Côtes-d'Armor departement
watershed	watershed	polygon	Watershed name's Sterenn
	subwaterhed	polygon	Sub watershed in Sterenn
	wsmunicipalities	polygon	Municipalities crossing Sterenn watershed
roads_waternetwork	roads	line	Roads network in Sterenn watershed
	waternetwork	line	Ditches and rivers in Sterenn watershed
landcover_hedgerow	landcover2000	polygon	Land cover classification in 2000
	hedgerow	line	Hedges network in Sterenn watershed
topographicmap	F018_024.TIF	Raster image	Topographic map (tif + world file)
	F018_025.TIF		Topographic map (tif + world file)
aerialphoto1998	photo2_ecw.ecw		Aerial photography (ecw)
	photo3_ecw.ecw		
pop22-5499			Population by municipality 1954 to 1999 in Côtes-d'Armor (Bretagne)
dataRGA			Farming evolution by municipality

# Using OpenJUMP

## 1 Loading and displaying data

### 1.1 Loading and organizing data

When OpenJump starts, you will see a main window named "project" (figure 3). A "project" is not only a window which allows data visualisation and organisation but it's a file too, which must contain all the work you do with OpenJump (.jmp extension).

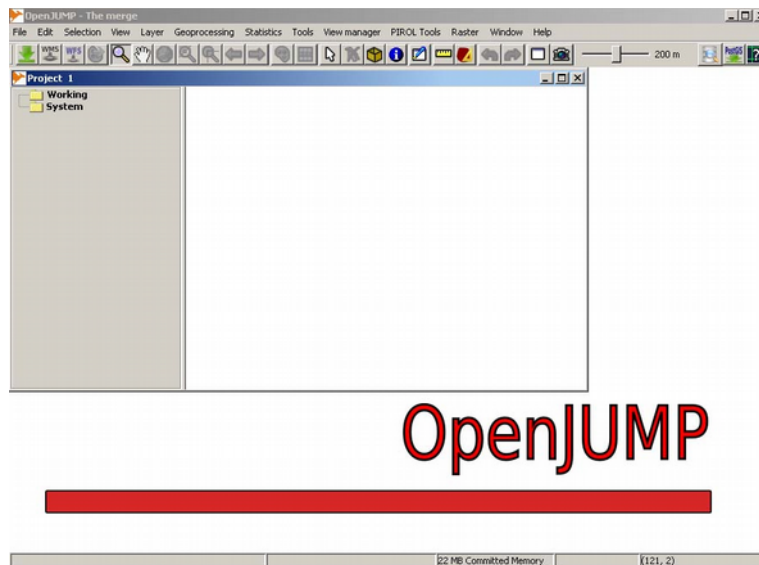


Figure 3

First, you will load some vectorial layers. So from the File Menu, choose "Select datasets". You have to select :

- frenchprovinces.shp,
- bzhdepartements.shp,
- municipalities22.shp,

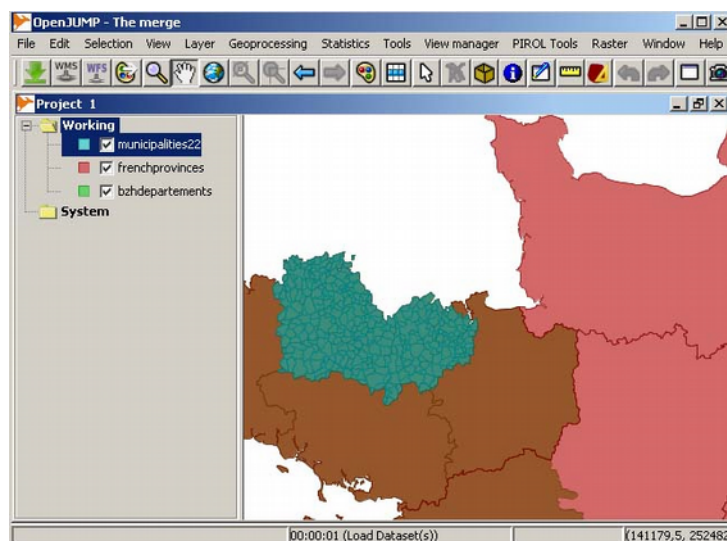







Figure 4

Loaded layers are listed in a table of contents, to the left of the visualisation window (figure 4). We will call it *Layer View*.

There are many ways to move in the *Layer View* and to zoom in to or out of the areas shown in it. So, we will ask you to use navigation tools (directly from the toolbox, from the View Menu or by a right-click in the table of contents).

zoom in/out,	
zoom to full extent	
zoom previous/next	
zoom to fence	
zoom to selected items	
zoom to scale	Turn on View > Zoom to scale
zoom to layer or layers	Right-click on a selected layer

## 1.2 Modifying layer properties

To modify layer styles, you can use the Styles dialog (figure 5).

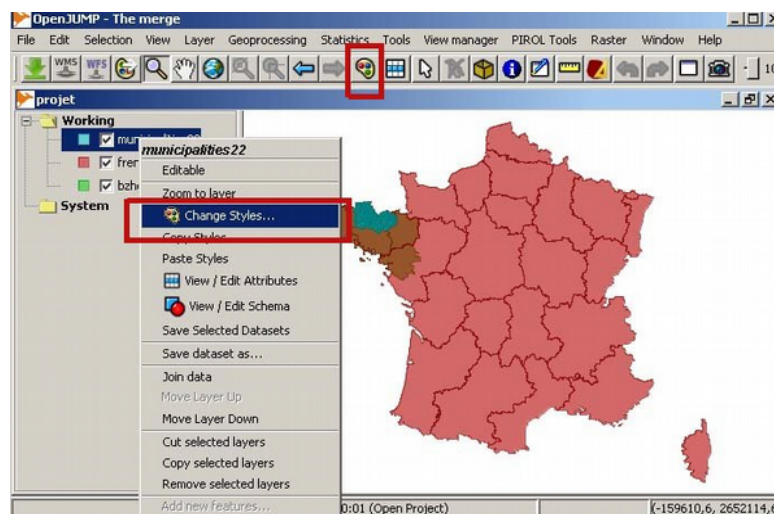


Figure 5

The **Rendering** Tab (figure 6) enables you to change general properties (colour, transparency, line width and pattern, vertices visibility).

The **Scale** Tab (figure 7) allows you to customize the scale display. So you can define that a layer can be shown only between a smallest and a largest scale. To know the current scale, you must select "show scale" from the View Menu.

The **Labels** Tab (figure 8) enables you to define labelling options (font : type, colour and height, placement). You have to know especially what attributes you want to label on the layer.



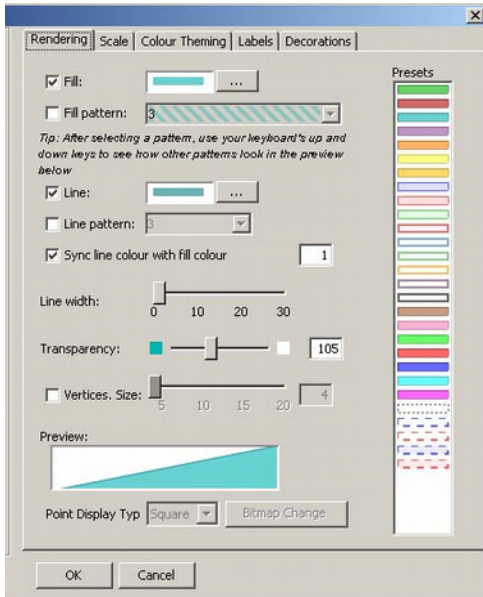


Figure 6

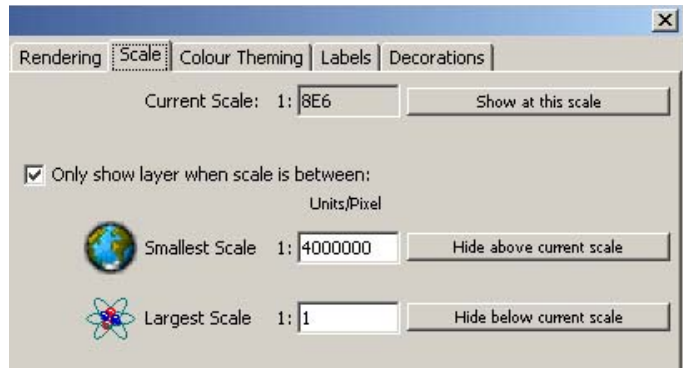


Figure 7

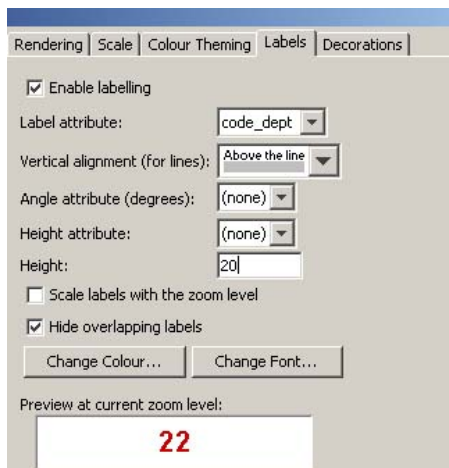


Figure 8

**Exercise** (figure 9)

Symbolise frenchprovinces  
 Uncheck fill  
 Set gray lines width=7

Symbolise bzhdepartements  
 Uncheck fill  
 Set black lines width=2  
 Label using field name\_dept

Symbolise municipalities22  
 Set presets style yellow

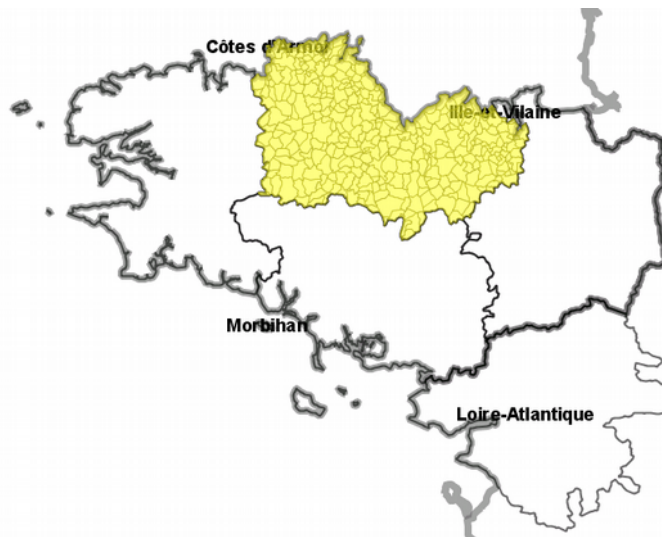


Figure 9

Then, from the File Menu, you will load some raster files :

- photo2\_ecw.ecw
- photo3\_ecw.ecw
- F018\_024.TIF
- F018\_025.TIF

Raster files should be listed in a new category (named *Fond*) in the table of contents. In the *Layer View*, to have a better visibility, you can create, rename and move the categories in the table of contents and order as you want (figures 10, 11 and 12).

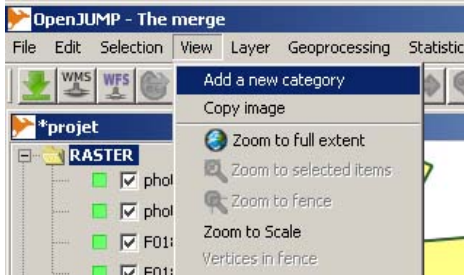


Figure 10

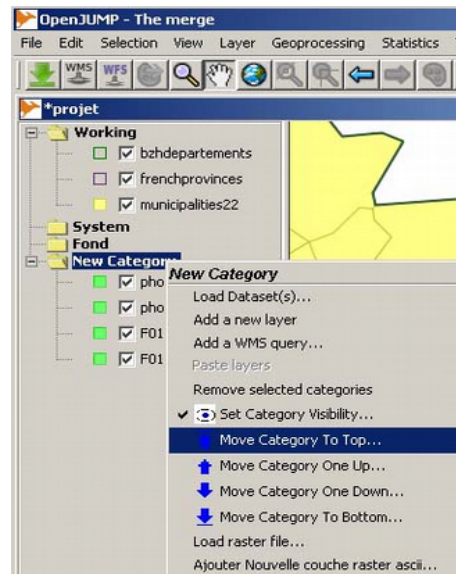


Figure 11

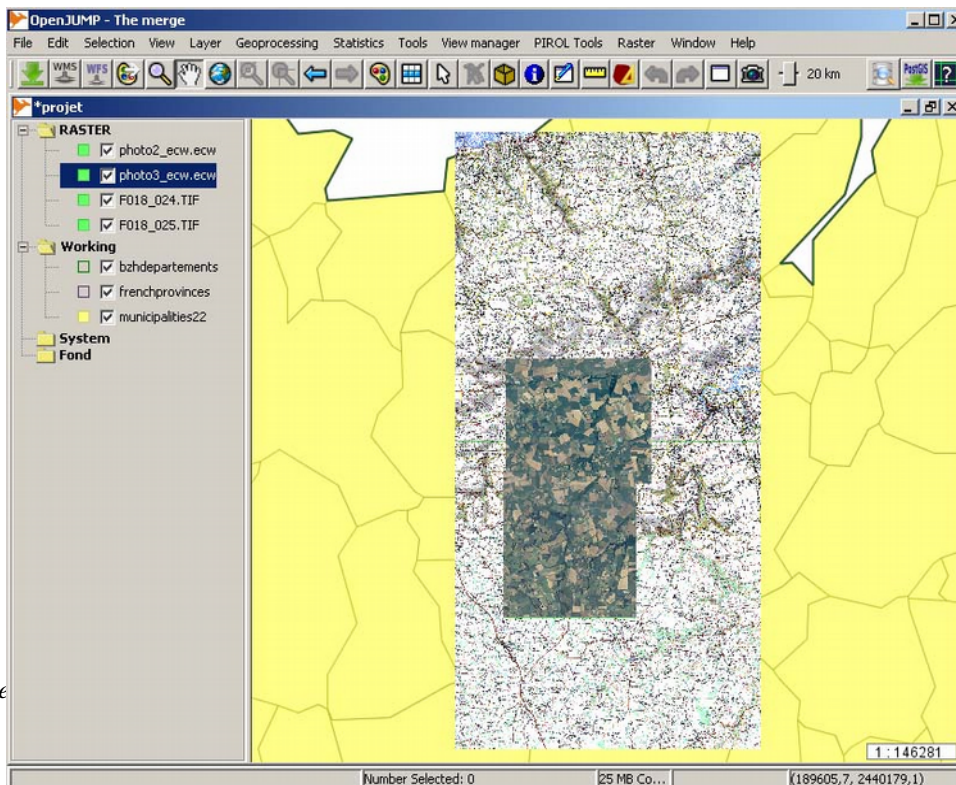


Figure 12

### Exercise : customize the raster scales

For photo2\_ecw.ecw and photo3\_ecw.ecw, smallest scale = 12 000  
For F018\_024.TIF and F018\_025.TIF, smallest scale = 15 000

To help you, you can display the scale in numeric value. Turn on View>Show scale

## 2 Working with alphanumeric data

### 2.1 Viewing attributes

In a layer, each feature has alphanumeric attributes. If you want to inspect the attributes of layer, you can use the toolbox or choose "View/Edit attributes" after a click-right in the table of contents (figure 13).

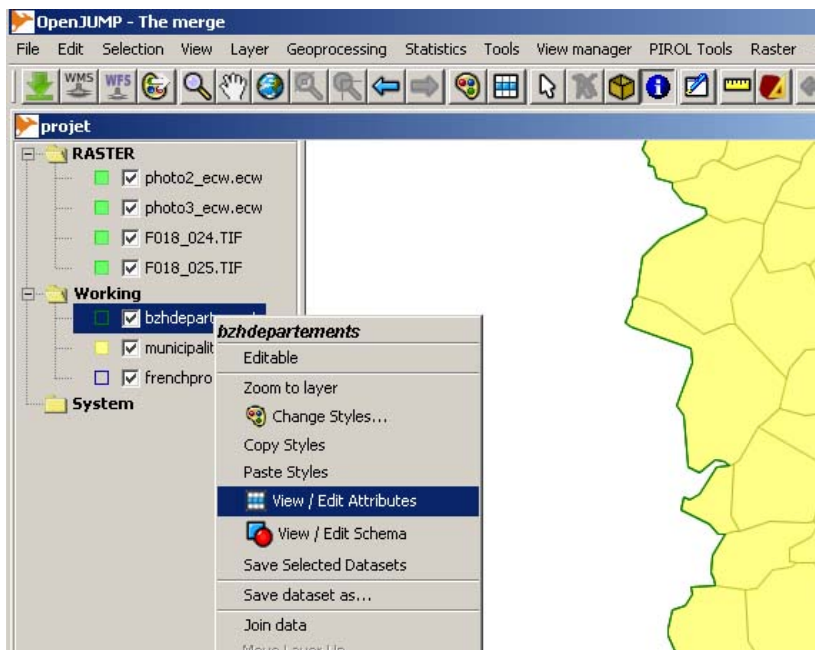



Figure 13

You will see a new window named "Attributes" (figure 14).



Figure 14

Another way to reach specific attributes is available by selecting one or many features directly in the *Layer View* with the Info tool 

## 2.2 Importing and joining alphanumeric data

There, we want to join demographic data with the layer named "municipalities22".

Open pop22-5499.ods with OpenOffice. Turn on File> Save as. Choose text CSV. Set values as figure 15.

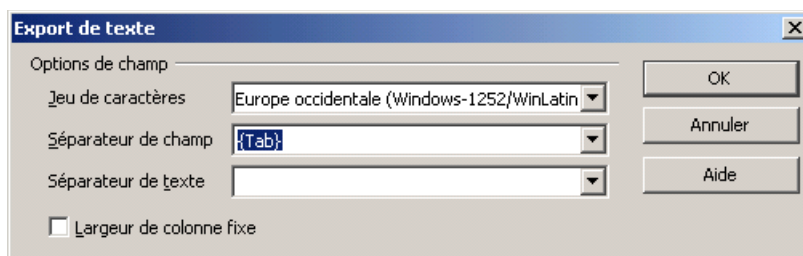


Figure 15

Right-click on "municipalities22" in the table of contents, and choose "join data" (figure 16).

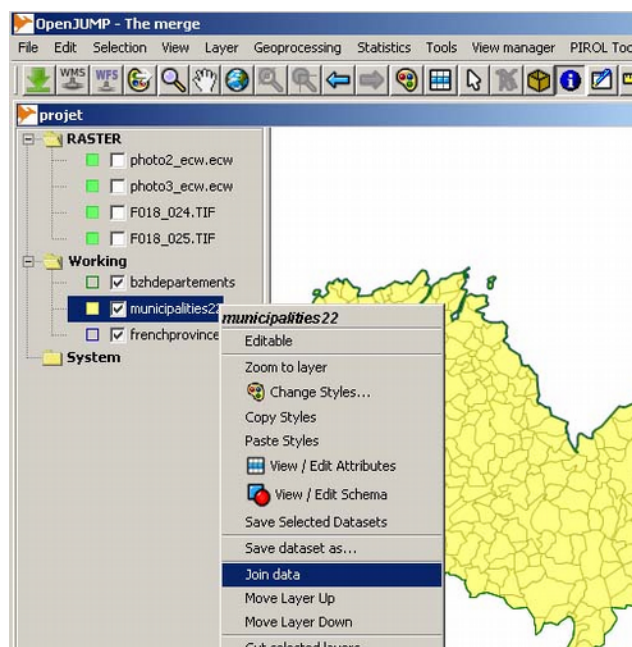


Figure 16

In the dialog that appears, All Files select file named "pop22-5499.csv".

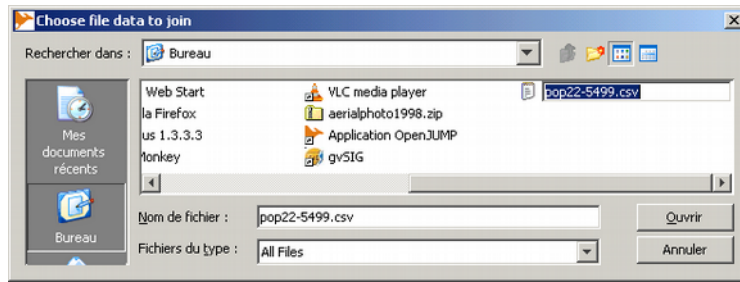
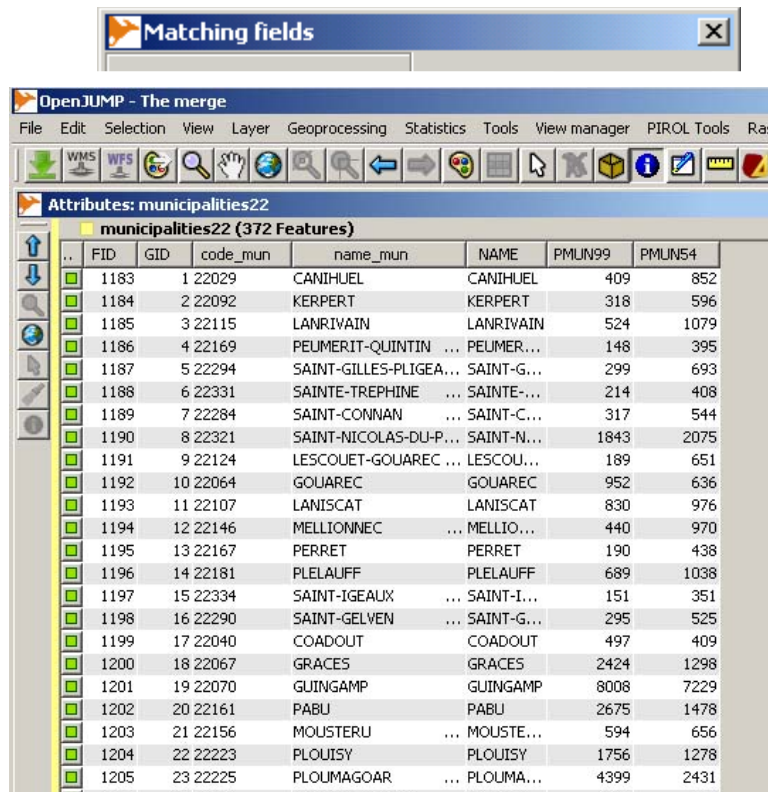


Figure 17

Click the first dropdown arrow and click the field name in the layer that the join will be based on (Figure 18).

Click the second dropdown arrow and click the field in the table to base the join on (Figure 18).

Click OK



The attributes of the table are appended to the layer's attribute table (figure 19).

Figure 19

To keep the result, you must save the layer.

## 2.3 Modifying layer schema

With OpenJUMP you can easily add or remove fields from a table if the layer is editable. Right-click on the layer municipalities22 and check "Editable".

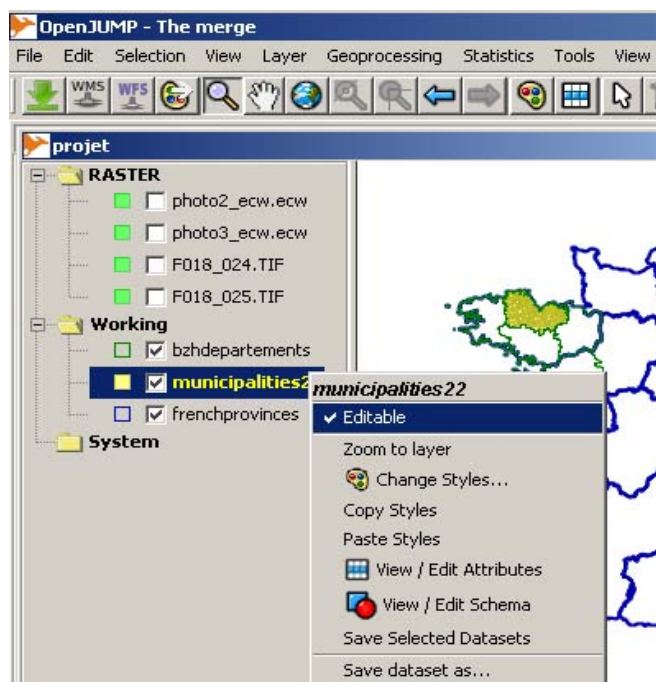
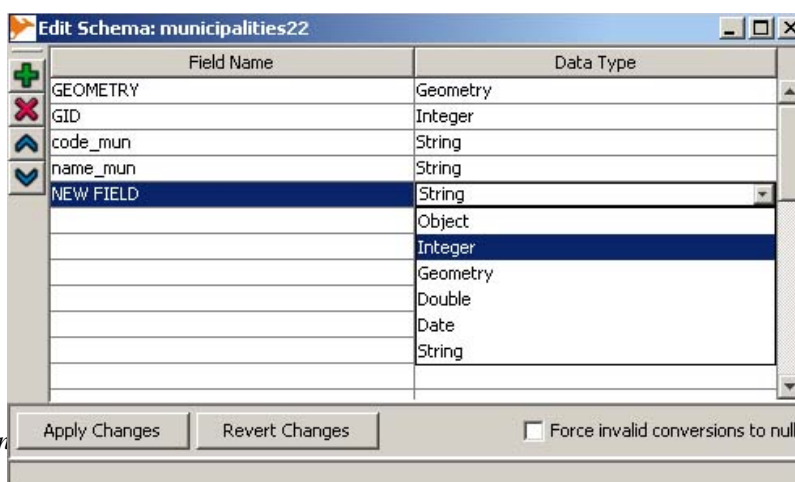


Figure 20

Right-click on the layer municipalities22 and select "View/Edit Schema". A new window appears. All fields are listed and described. By this way, we can know the data type of each field (string, numeric, boolean, geometry ...)

In the Edit Schema, click on the lasted row to add a new field "area". Click the Data Type dropdown arrow and click the field type. Click Apply Changes (figure 21).



Erwan Bocher, Arr

Figure 21

Then, it's possible to populate a field.

In this part, we will ask you to calculate areas of "municipalities22" features.

Turn on Statistic > Calculate areas and lengths. Select what kind of operation you want to do and specify the right field (figure 22).



Figure 22

The results are expressed as m<sup>2</sup>. You have to convert their as km<sup>2</sup>. The Pirol attribute tools lets you perform simple calculations on all records.

Turn on Pirol Tools>Attribute tools>Editeur de formules.

Set name AREA\_KM2

Click type DOUBLE

Set expression like

figure 23

Click OK

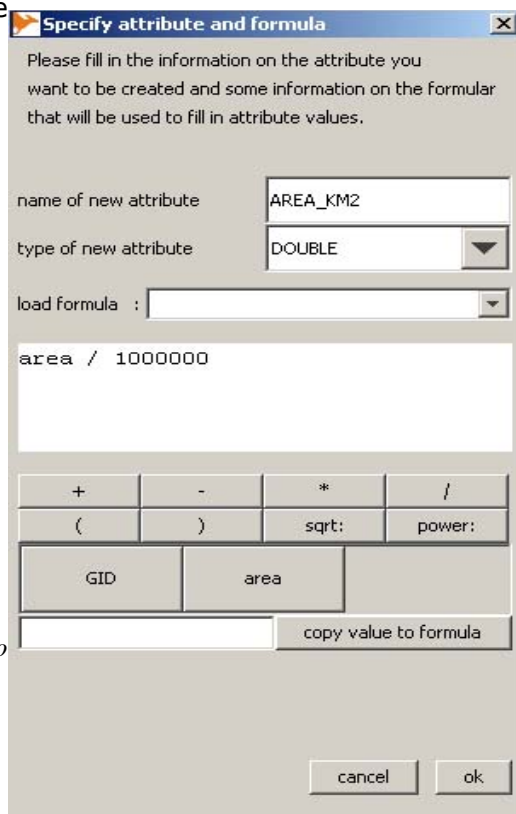


Figure 23

This tool will create automatically the new field (area\_km2) in the attribute table.

FID	GID	code_mun	name_mun	NAME	PMUN99	PMUN54	area_m2	area_km2
1520	372	22349	TREDREZ-LOCQU...	TREDREZ-LOCQUEMEAU	1250	944	1.0676194877870172E7	10.676194877870172
1521	371	22317	SAINT-MELOIR-D...	SAINT-MELOIR-DES-B...	233	278	6852398.19298923	6.85239819298923
1522	370	22352	TREFUMEL	TREFUMEL	246	314	5550355.525026679	5.550355525026679
1523	369	22308	SAINT-JUVAT	SAINT-JUVAT	648	855	1.812316485231605E7	18.12316485231605
1524	368	22306	SAINT-JUDOCE	SAINT-JUDOCE	443	482	1.0515069256707191E7	10.515069256707191
1525	367	22208	PLOUASNE	PLOUASNE	1358	1889	3.447812144284603E7	34.47812144284603
1526	366	22274	SAINT-ANDRE-DE...	SAINT-ANDRE-DES-EAUX	237	254	5925529.879997849	5.92552987999785
1527	365	22263	LE QUIOU	LE QUIOU	305	337	5514057.663902909	5.514057663902909
1528	364	22056	EVRAIN	EVRAIN	1473	1520	2.4037042553700924E7	24.037042553700925
1529	363	22035	LES CHAMPS-GER...	LES CHAMPS-GERAUX	960	1007	1.986553043524301E7	19.86553043524301
1530	362	22364	TRELIVAN	TRELIVAN	2170	627	1.1634704705414817E7	11.634704705414817
1531	361	22327	SAINT-SAMSON-S...	SAINT-SAMSON-SUR-...	1151	540	6022859.92059423	6.022859920594231
1532	360	22380	TREVRON	TREVRON	664	613	9726735.835973784	9.726735835973784
1533	359	22280	SAINT-CARNE	SAINT-CARNE	811	616	8828917.762877464	8.828917762877465
1534	358	22339	TADEN	TADEN	1741	1034	2.0505404189522088E7	20.50540418952209
1535	357	22259	QUEVERT	QUEVERT	3118	1155	1.258712201368554E7	12.587122013685555
1536	356	22213	PLOUER-SUR-RA...	PLOUER-SUR-RANCE	2723	2278	1.955709573626717E7	19.557095736267172
1537	355	22082	LE HINGLE	LE HINGLE	680	383	3128461.5569962338	3.1284615569962337
1538	354	22026	CALORGUEN	CALORGUEN	523	516	8327255.093175203	8.327255093175202
1539	353	22021	BRUSVILY	BRUSVILY	794	643	1.23113166164999E7	12.311316616499901
1540	352	22008	BOBITAL	BOBITAL	865	323	5043883.223474801	5.043883223474801
1541	351	22003	AUCALEUC	AUCALEUC	629	326	6367302.495443866	6.367302495443866
1542	350	22050	DINAN	DINAN	10885	11550	3848179.3760320693	3.8481793760320695
1543	349	22385	LA VICOMTE-SUR...	LA VICOMTE-SUR-RANCE	770	752	4635148.367544368	4.635148367544368
1544	348	22390	SAINT-HELENI	SAINT-HELENI	1021	871	68742762342595E7	16.8742762342595E7

Figure 24

**Exercise** (figure 25)

Calculate population densities in 1999 and 1954.

PMUN99 and PMUM 54= population, numbers are in thousands.

Population density (habitants/square kilometres)

area_km2	density99	density54
10.676194877870172	117.08291336935267	88.42101617653513
6.85239819298923	34.00269415726381	40.56973809321605
5.550355525026679	44.321485153659154	56.572952594508...
18.12316485231605	35.75534434964811	47.17719046134126
10.515069256707191	42.130012573852134	45.83897530608743
34.47812144284603	39.38729673109193	54.78836783875747
5.92552987999785	39.99642307095851	42.86536481022557
5.514057663902909	55.313168376283855	61.116517189533...
24.037042553700925	61.2804173687002	63.23573279051209
19.86553043524301	48.32491149075409	50.69081861582226
11.634704705414817	186.51096481976694	53.890495364974...
6.022859920594231	191.10522495539618	89.65840267238396
9.726735835973784	68.26545011577609	63.02217006170292

Erwan Bocher, Arnau...

Figure 25



### 3 Thematic maps

First, create a new project. In this project, copy the three vectorial layers (copy-paste).

Turn on View Manager > Mosaic.

Turn on View, select "active synchronisation" a new button "Synchronize" must be appear in the menu tool bar. Click on. Zoom in to a view. Views are synchronized (figure 26).

Don't forget, you should have the same vectorial layers in both projects.

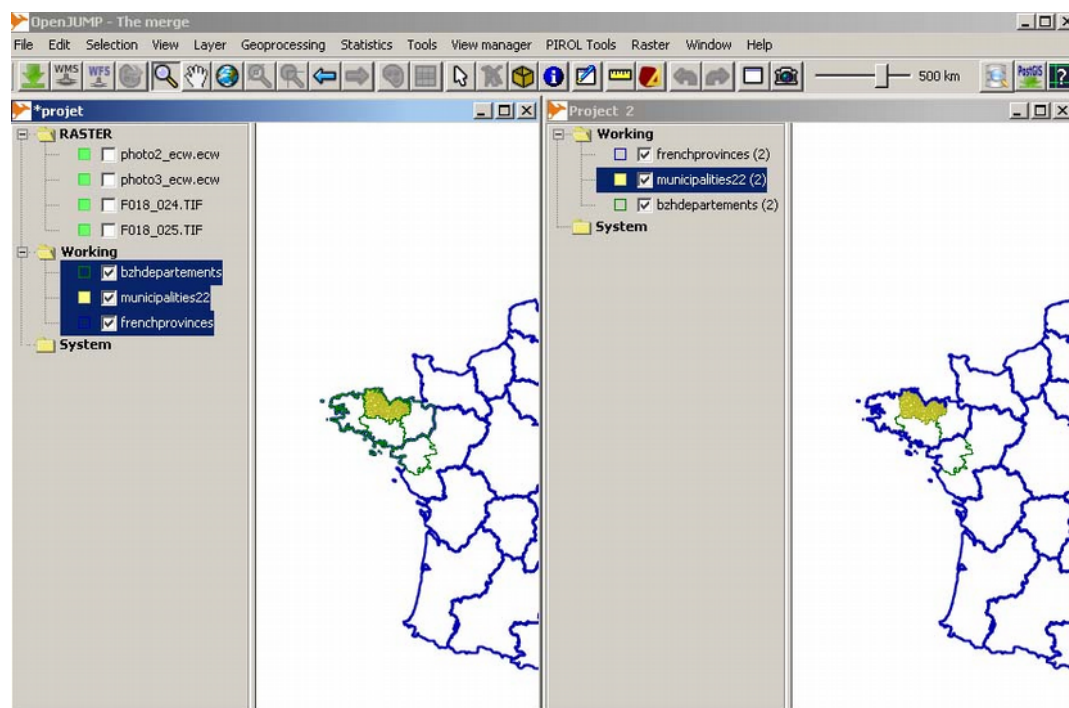


Figure 26

In each project, you will colour-theme municipalities layer by one attribute (figure 28).

- project 1 : density 1954
- project 2 : density 1999

Open the "change styles" window.

The Colour theming tab enables you to choose the classification method, the number of ranges and the attribute to colour theme. Then it's possible to modify the colour scheme (figure 27).

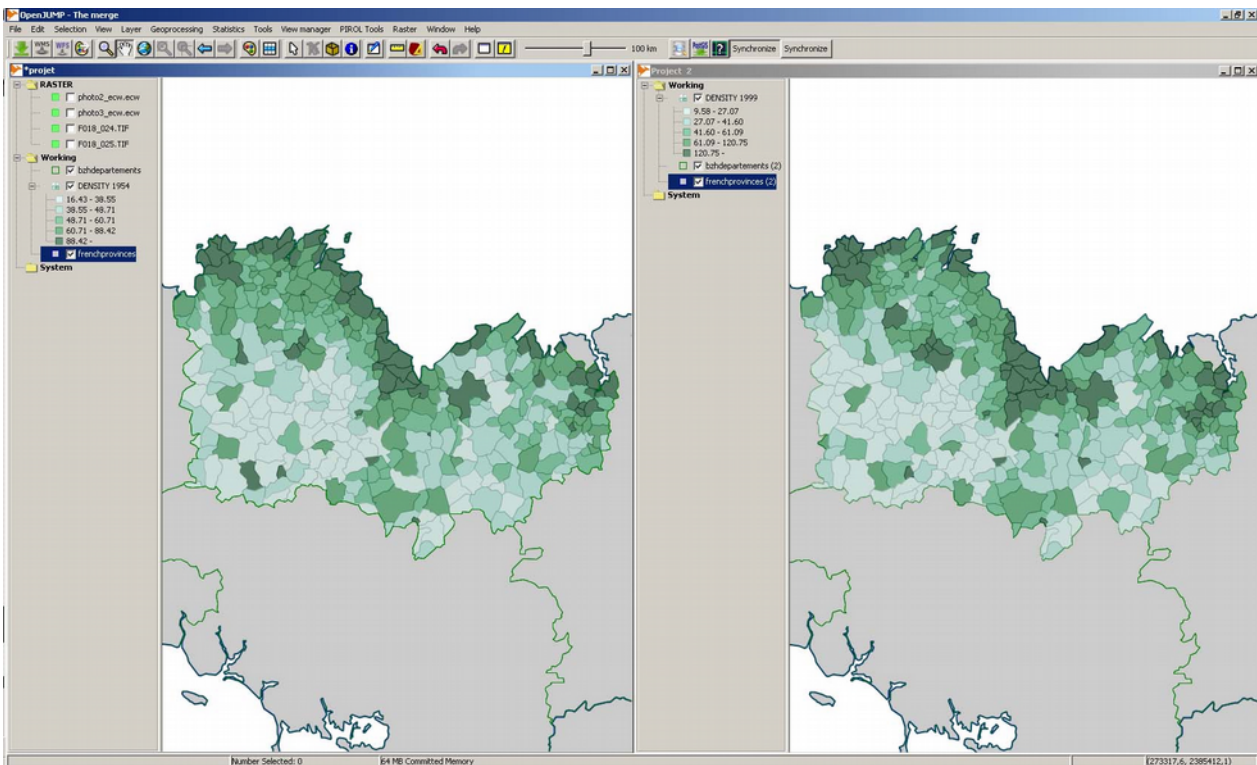
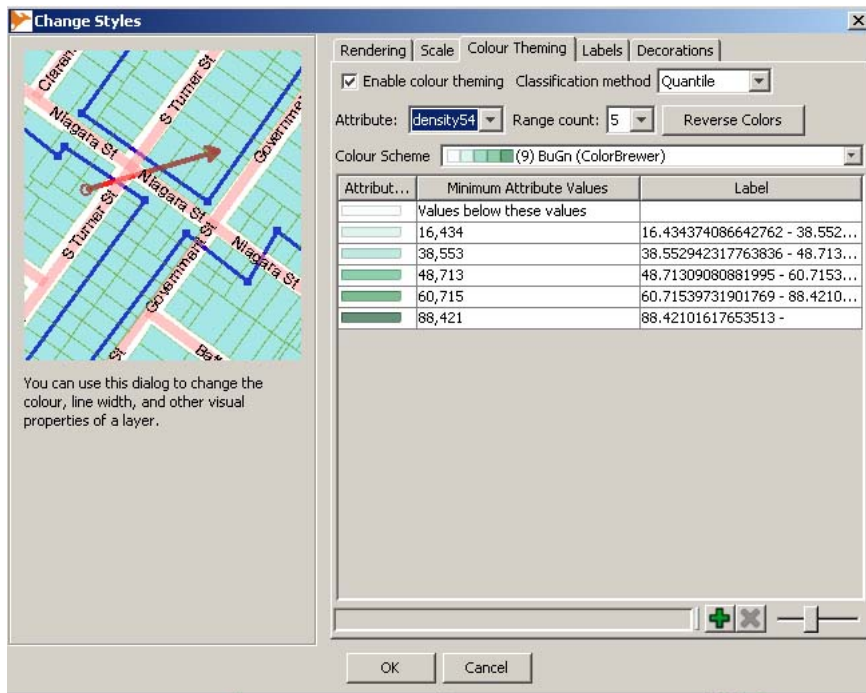


Figure 28

## 4 Exporting views

For each projects add a scale bar. Turn on View>Scale Bar.

You can export a layout and its legend to svg or png files. So you can complete, modify their, in another programs (Inkscape for example)

From File Menu, select "save image in a svg format" or "save image as" (figure 29). Right-click on the layer in the table of contents and choose "save legend as png".

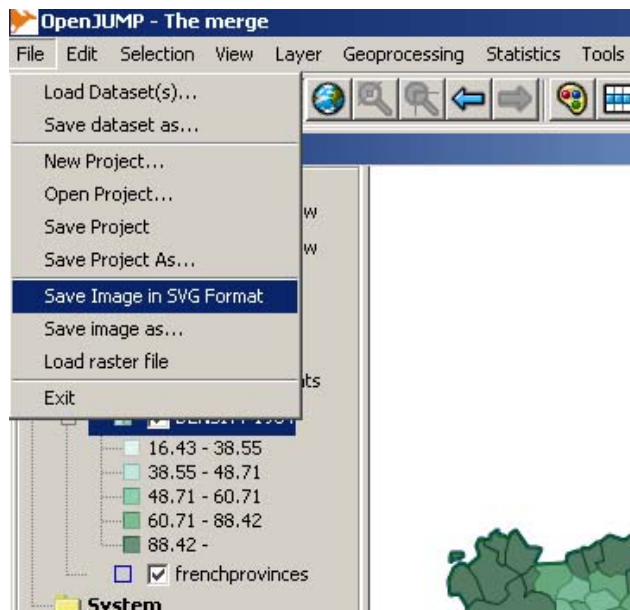


Figure 29

## 5 Querying

### data

#### 5.1 Simple attributes queries

Load datasets (figure 30) :

- landcover2000.shp,
- rivers.shp,
- roads.shp,
- hedgerow.shp,
- watershed.shp,
- subwatershed.shp,
- wsmunicipalities.shp.

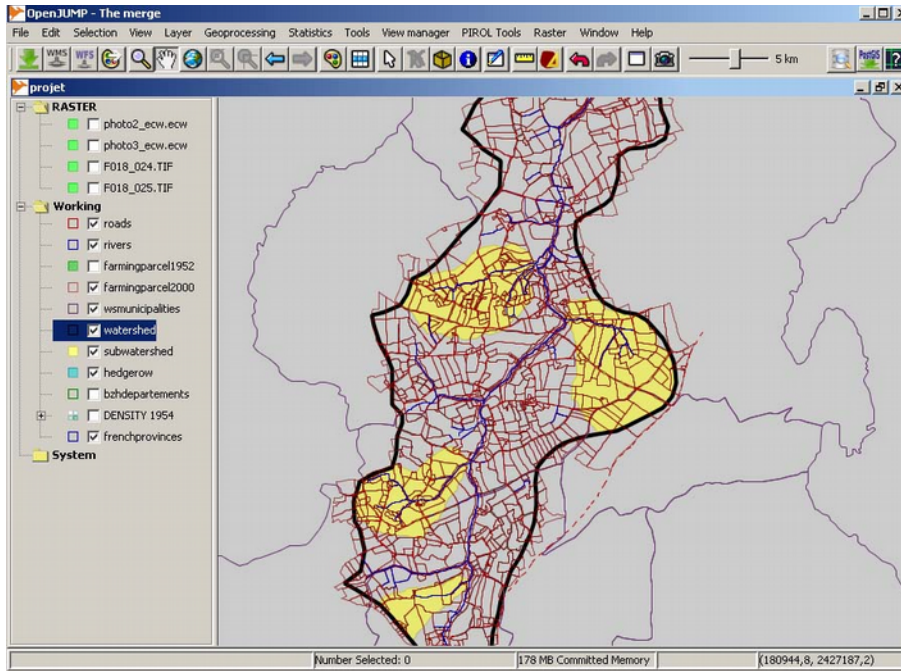


Figure 30

Select forest parcels (in 2000) with areas exceeding one hectare (10 000 square meters).

Turn on Selection>Queries>simple query. it will open a new dialog named "Query Builder" (figure 31).

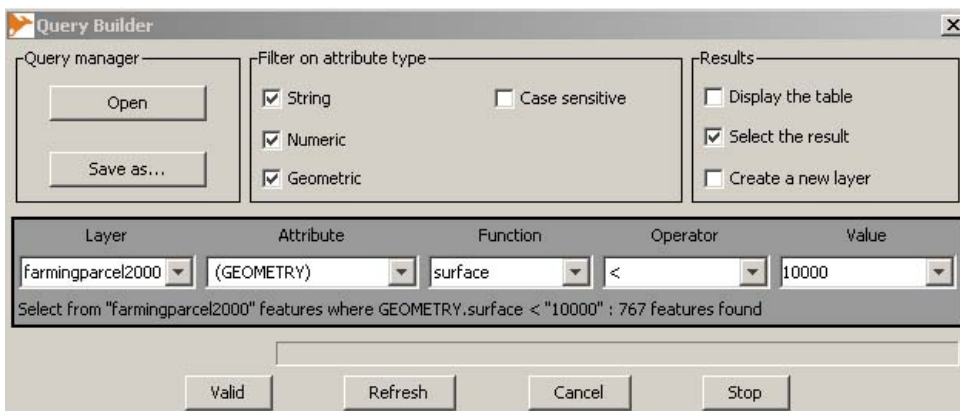


Figure 31

The Query Builder enables to save your queries and to choose what kind of result you want (features selection, table display or new layer)

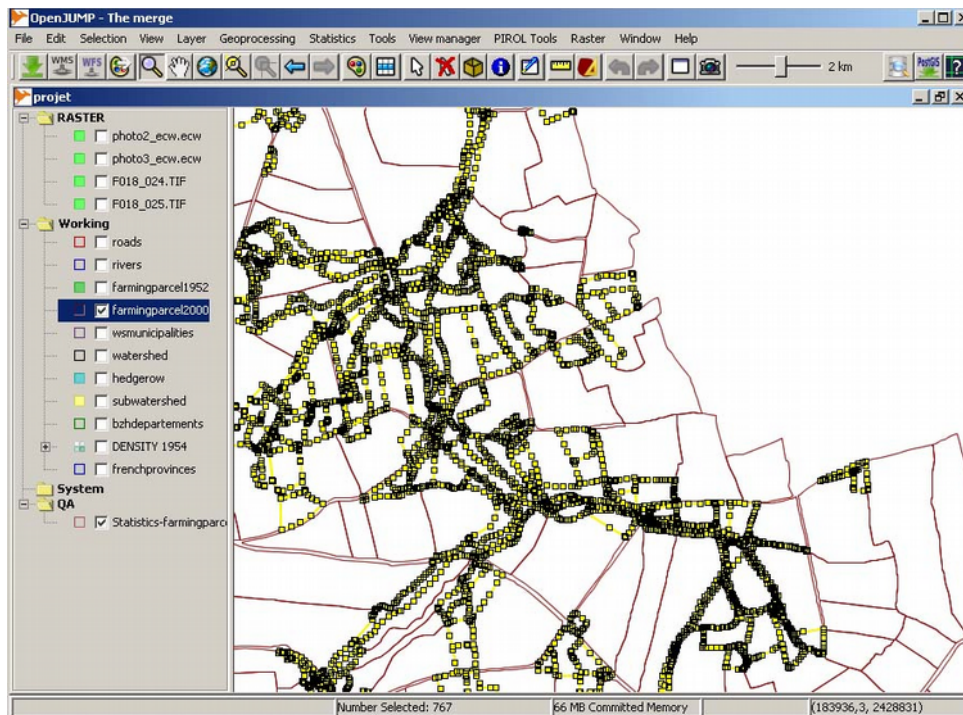


Figure 32

Select moisture value and replace it by "wetlands".

Go to Query Builder. Select moisture value in landcover2000.

Make layer editable.

Turn on Layer>Replace value (figures 33 et 34).



Figure 34

FID	type	runoff_wint	runoff_summ
2617	wetlands	0.05	0.05
2618	wetlands	0.05	0.05
2646	wetlands	0.05	0.05
2648	wetlands	0.05	0.05
2656	wetlands	0.05	0.05
2663	wetlands	0.05	0.05
2677	wetlands	0.05	0.05
2684	wetlands	0.05	0.05
2695	wetlands	0.05	0.05
2705	wetlands	0.05	0.05
2711	wetlands	0.05	0.05
2712	wetlands	0.05	0.05
2771	wetlands	0.05	0.05
2772	wetlands	0.05	0.05
2784	wetlands	0.05	0.05
2786	wetlands	0.05	0.05
2787	wetlands	0.05	0.05

Figure 33

## 5.2 Spatial queries

What are the parcels within 100 meters of roads ?

Go to Query Builder. Set (figure 35)

Layer = landcover2000

Attribute = (GEOMETRY)

Operator = is within (100.00)

Value = road

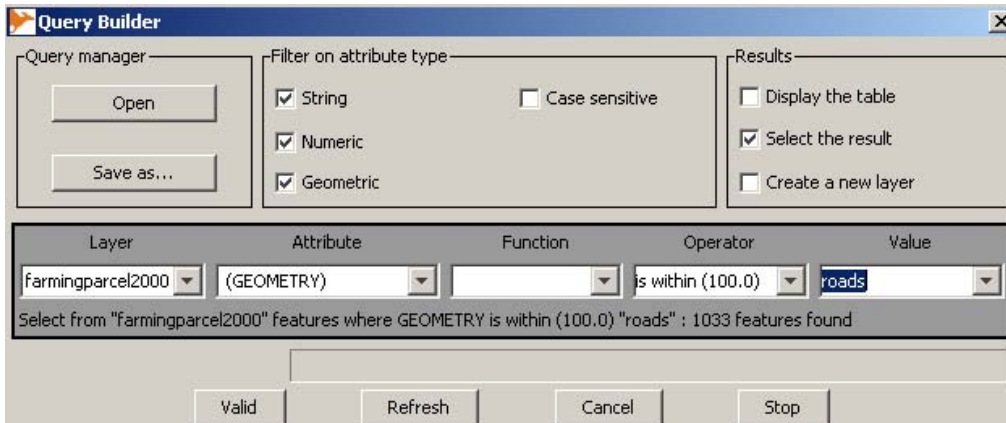


Figure 35

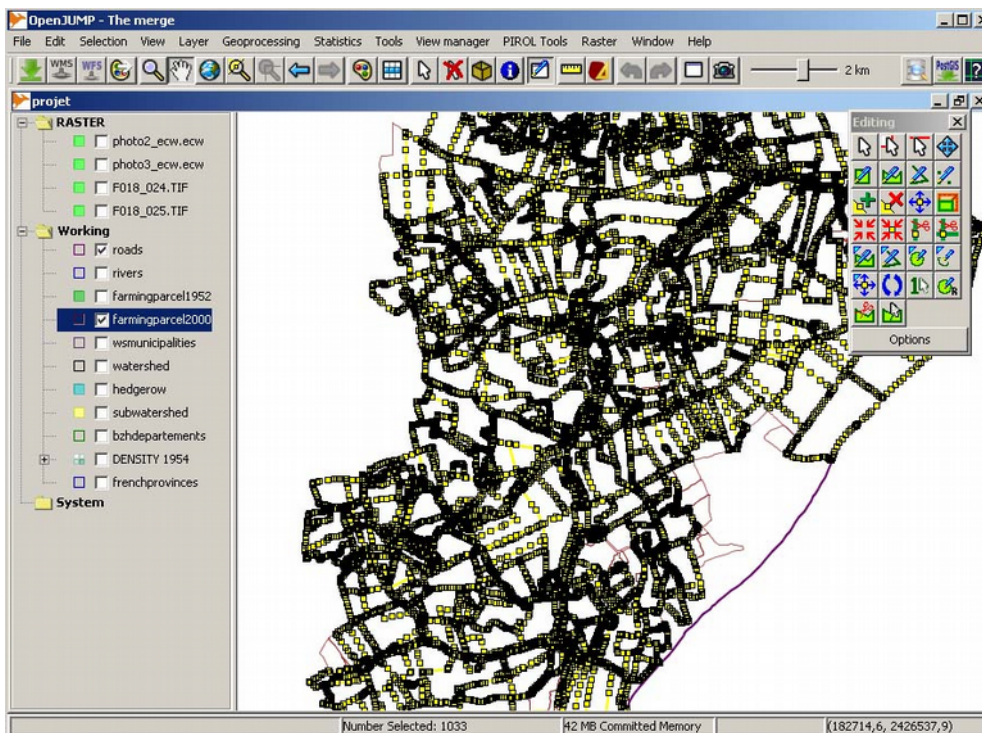


Figure 36

## **Exercises**

### **Exercise1**

What are the wetland parcels within 100 meters of municipal roads ?

### **Exercise2**

What are the hedges which touch grassland parcels ? Note their total length and display the result in a new layer.

# Using OpenJUMP in a SDI context

## 1 OpenJUMP and Postgis

### 1.1 Create a database and load postgis extension

Start pgAdmin tool to manage database.

Go to file Add server

**Set :**

Address : localhost

Description : Foss4g workshop

Service : « null »

Maintenance DB : template1

User name : admin

Password : foss4g

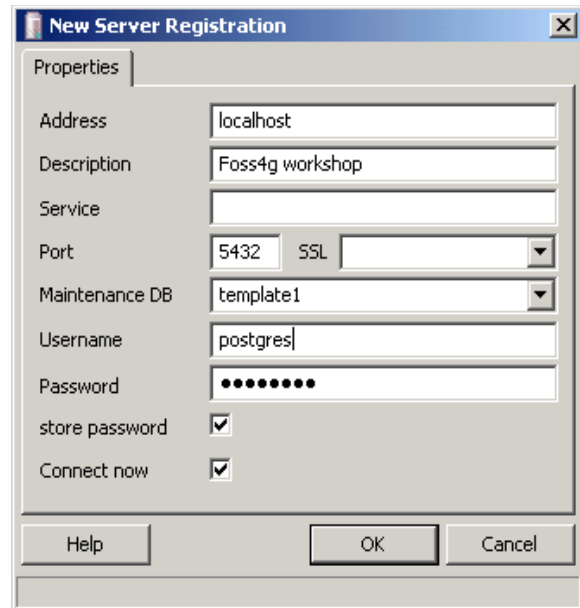


Figure 37

You should obtain this (figure 38).

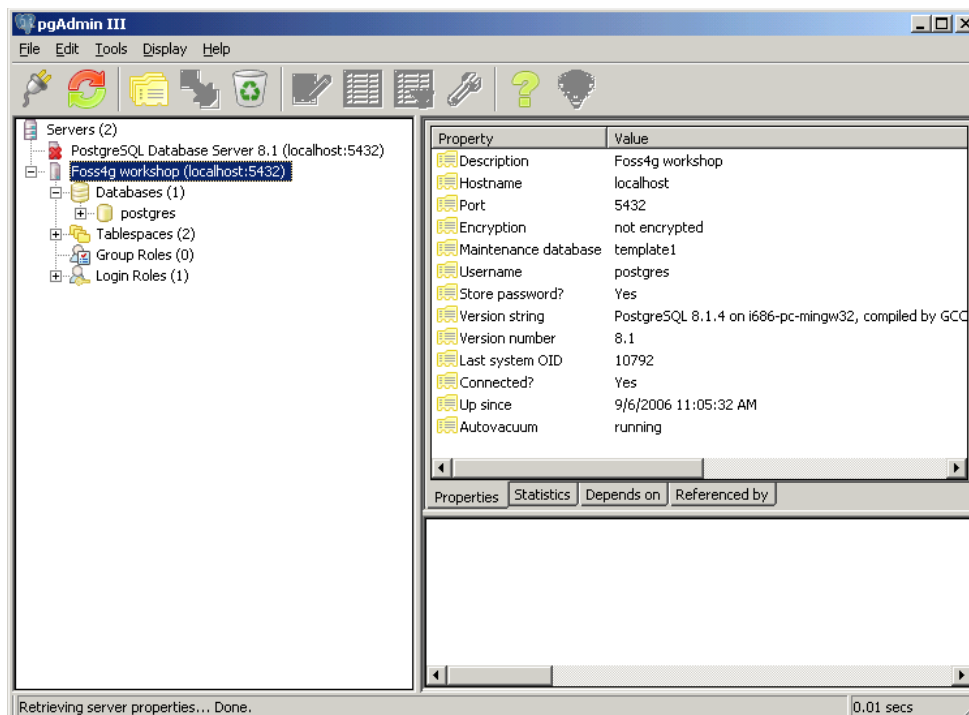


Figure 38



## Create a database (figure 39)

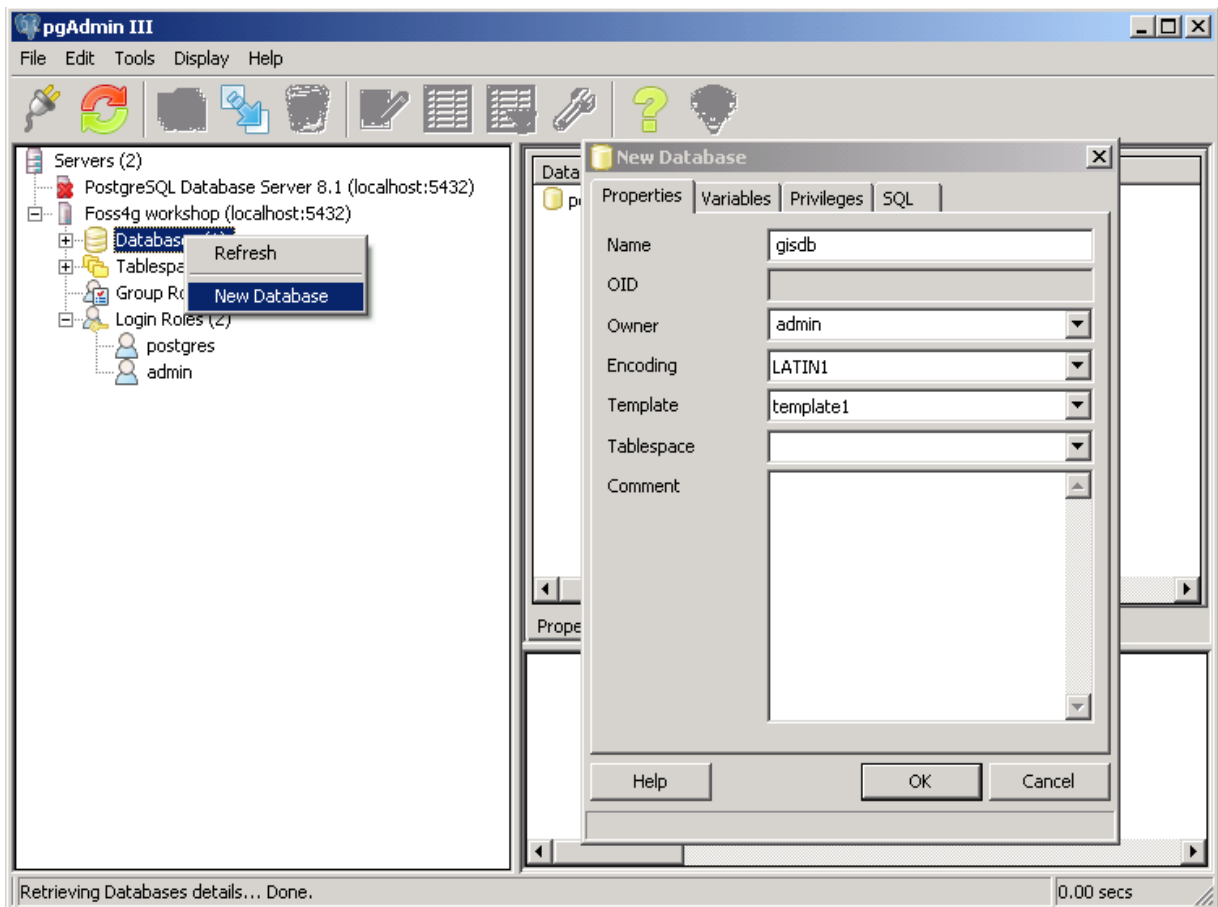


Figure 39

The PostGIS extension requires the PL/PgSQL procedural language for some of its functions.

If plpgsql doesn't exist. Right-click on language and set parameters like figure 40.

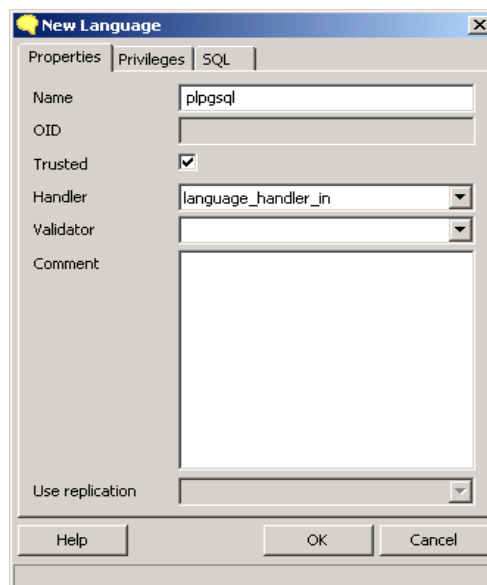



Figure 40

Now, let's go to load postgis functions.

Open SQL query builder 

1. Load lwpostgis.sql and execute it (figure 41)

2. Load spatial\_ref\_sys.sql

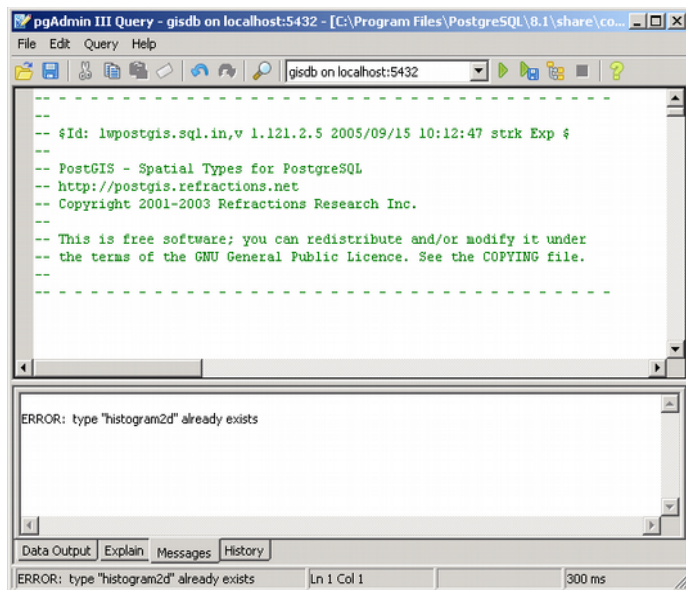


Figure 41

Now we are ready to import and to work with spatial data and features in PostgreSQL.

## 1.2 Using Postgis

### 1.2.1 Basic exercises

#### 1.2.1.1 Populating database

##### **Prepare data**

Create a new project. In this project :

Open shapefiles : "waternetnetwork.shp" "roads.shp" "watershed.shp"

Open rasters files : "photo2\_ecw.ecw" "photo3\_ecw.ecw"

For each layers

Open layer schema and change geometry name by the\_geom.

Add a gid column. Right-click on a layer and select « Add gid column ».

Change SRID add code for french projection : EPSG:27582 - NTF(Paris)/Lambert II (étendu)

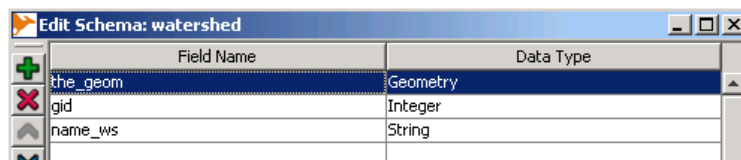


Figure 43

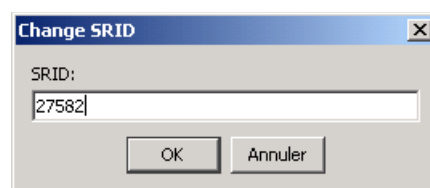


Figure 42

Create a new project. In this project :

Open shapefiles : "waternetnetwork.shp" "roads.shp" "watershed.shp"

Open rasters files : "photo2\_ecw.ecw" "photo3\_ecw.ecw"

Right-click on a layer and select  
« Change SRID ».

### **Import shaphefile in gisdb**

Select the layer to be imported

Turn on File>Save dataset as

In the combobox Format choose  
Postgis Table.

Figure 44 show parameters and  
the option to create a new table  
name watershed in gisdb  
database

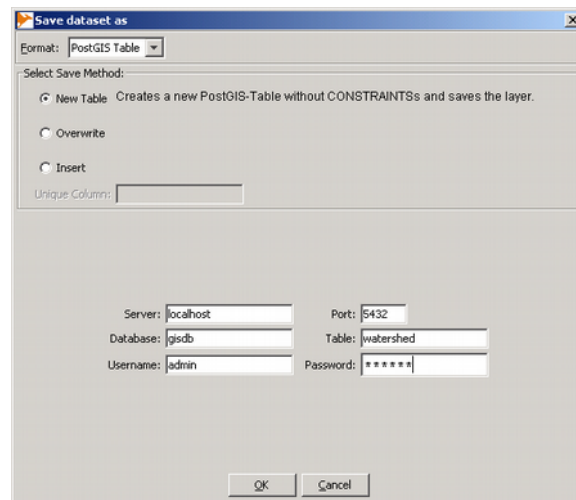


Figure 44

### **1.2.1.2 Display postgis layer in OpenJUMP**

Turn on File>Load dataset as

In the combo box Format choose  
Postgis Table.

Set :

Connexion name : foss4g

Server : localhost

Database name : gisdb

User : admin

Password :foss4g

You can save the parameters by  
using the save icon (Figure 45).

Click on the button Connection

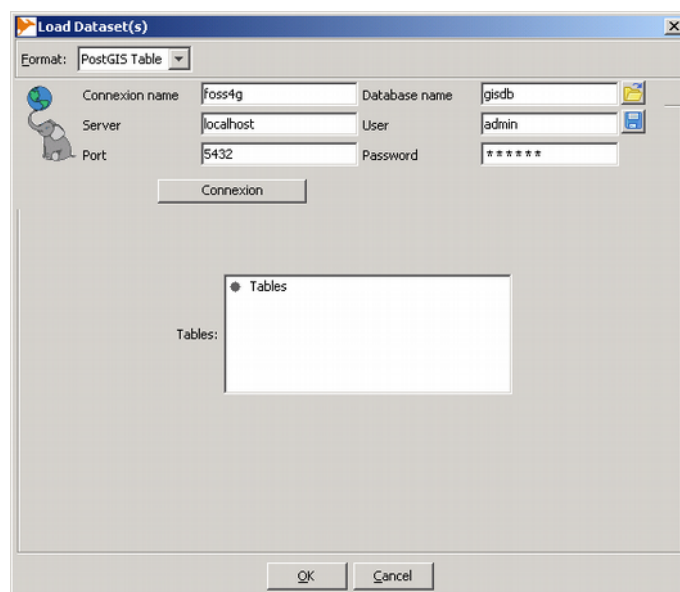


Figure 45

Figure 46 : The tables contained in the database are listed in a tree.

Select roads table and click OK.

The layer is loading in OpenJUMP.

Open waternetwork, roads and watershed layers.

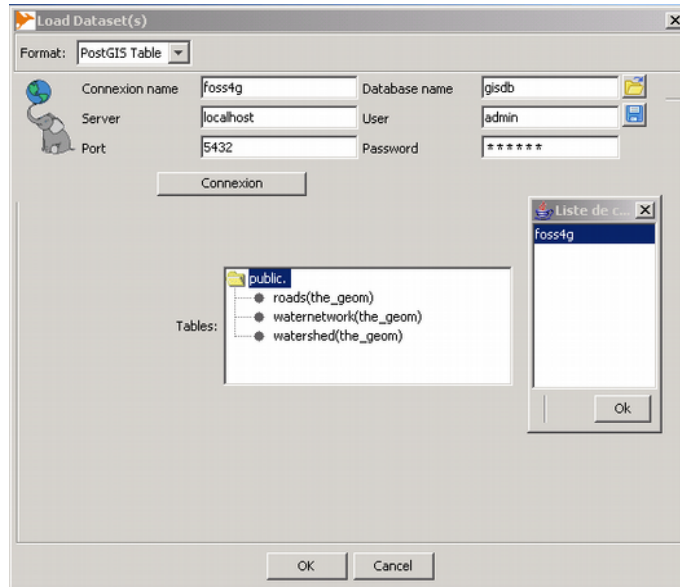


Figure 46

### 1.2.1.3 Editing, modifying and creating a postgis layer

Open rasters files : F018\_024.TIF and F018\_025.TIF.

Modify watershed style to display topographic maps.

Extract in another layer, the features waternetwork where the column type\_axe is equal to river value. Name this layer rivers.

Symbolize the new layer with a 4 thick blue line (Figure 47).

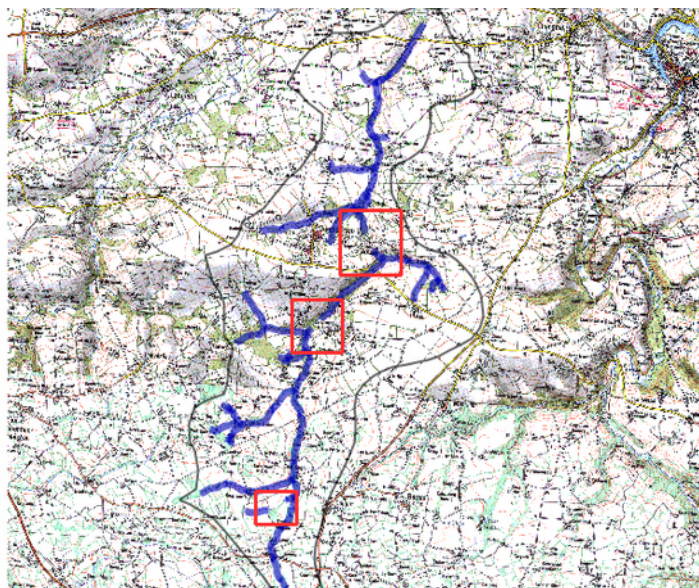



Figure 47  
27/41


Here you can show that features rivers aren't continuous. There are three breaks in the network (see red boxes on figure 47). The aim is now to complete the river network by drawing new features.


Zoom to the north red box


Make rivers editable


Use the draw line tool to create a new line 

If necessary reshape the line

by adding, 

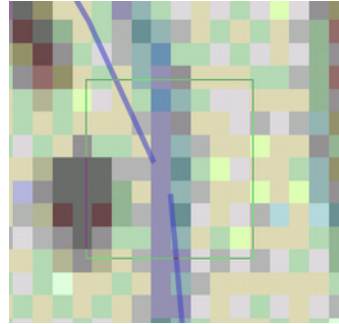
moving 

or deleting vertices. 

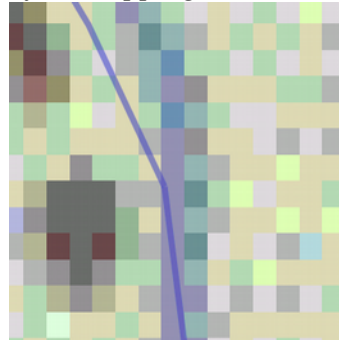
Zoom in to control snapping error  
Perform vertex by using snap box. 

To use it draw a rectangle across the vertex

*Before snapping*



*After snapping*



Set the attributes of the new features

Use the Feature Info tool to display its attributes

Put the value "river" into the type\_axe column  
(figure 48)

To do for each new features.

Don't forget to upload the gid column and save  
rivers in to postgis.

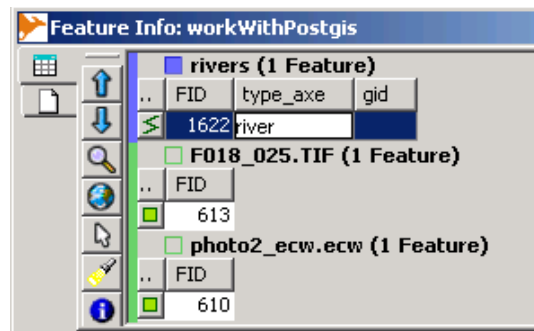


Figure 48

#### 1.2.1.4 Filtering a postgis layer

To filter a postgis layer and display results in OpenJUMP you can use ExecuteQueryDialog plugin, a small GUI to run sql syntax (Figure 49).

Turn on Layer>Ajouter une requête Postgis"or use this button



In the new window click on



Set postgis parameters and press



(figure 50)

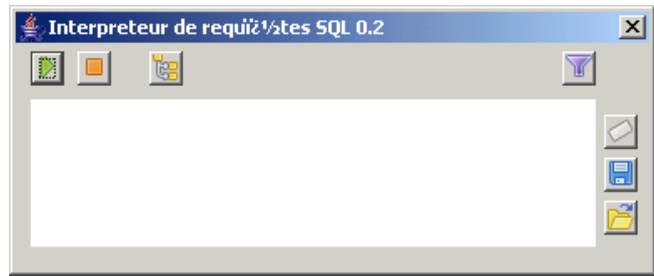


Figure 49

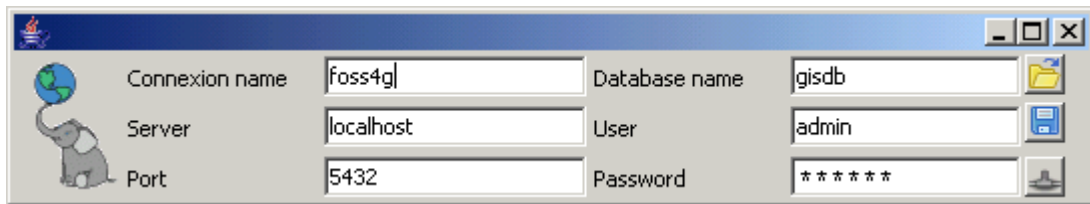


Figure 50

**Warning** there isn't any message to inform you about the connection. To test if the connection is activate close the window and click on

If you show a window like figure 7, your connection is right. Otherwise, you should repeat it again.

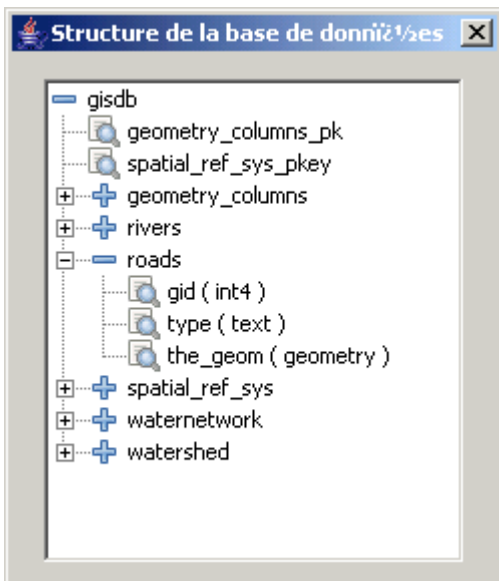


Figure 51

This window show the public schema of the database gisdb. By clicking on a tree node you can display :


- the name of the tables,
- the name of the columns and their format.

Filtering roads layer to display road where type is equal to municipality

You must use this syntax :


```
SELECT asewkt(<geometry column>) as the_geom, <column list> FROM <table> WHERE <column name>=<value>
```

So here (figure 52):

Press  
to execute the query 

A new category is created and a  
new layer appears.

To keep it, turn on File>Save data  
set as> Postgis Table

To save the query press 

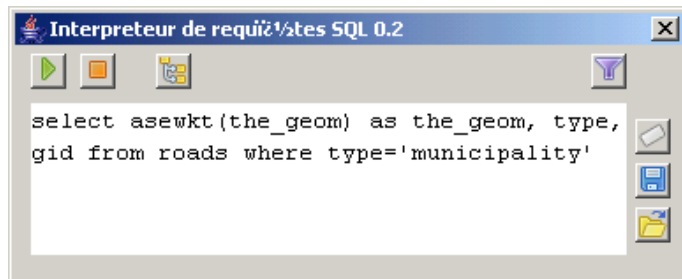


Figure 52

### Filtering roads layer by length

```
SELECT asewkt(the_geom) as the_geom, type, gid FROM roads WHERE length(the_geom) <500
```

Unit is metric (meters).

### **Exercise**

Find in landcover2000 features which are

- grassland and
- area greater than 2 hectares

The result must be display in OpenJUMP and layer must contains three columns :  
gid, type and area.

Notes :

function area is area(<geometry column>)  
10 000 square meters is equal to 1 hectare

Don't forget that SRID value is 27582 and unit is metric

## **1.2.2 Advanced exercises**



### 1.2.2.1 Configuring iSQLViewer

With the iSQLViewer an open source client for database, you can use more complex SQL expressions like CREATE, UPDATE... Moreover, you can use successive queries and bookmark them.

Turn on Layer>iSQLViewer for OpenJUMP (figure 53).

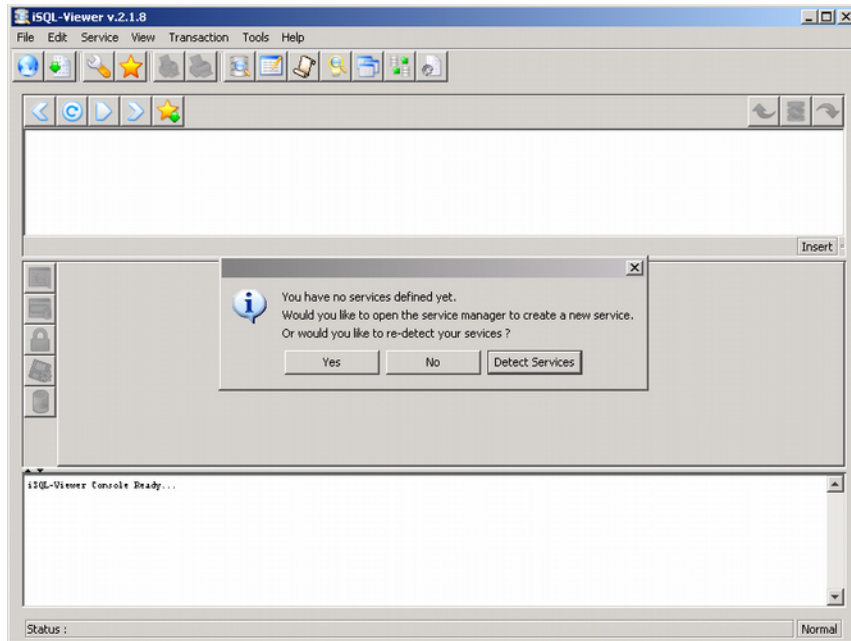


Figure 53

Click on Yes.

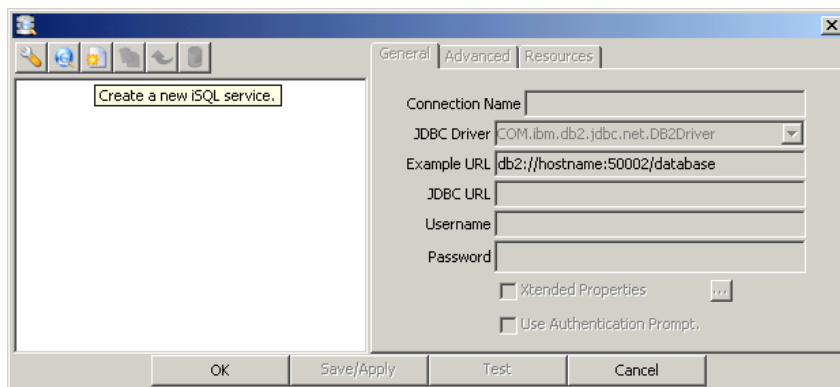


Figure 54

Click on Create a new iSQL service. Select Local Service.

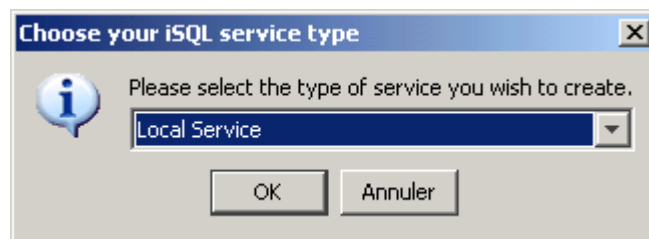


Figure 55

Set parameters as

showing on figure 56. Click on Save/apply (figure 57).

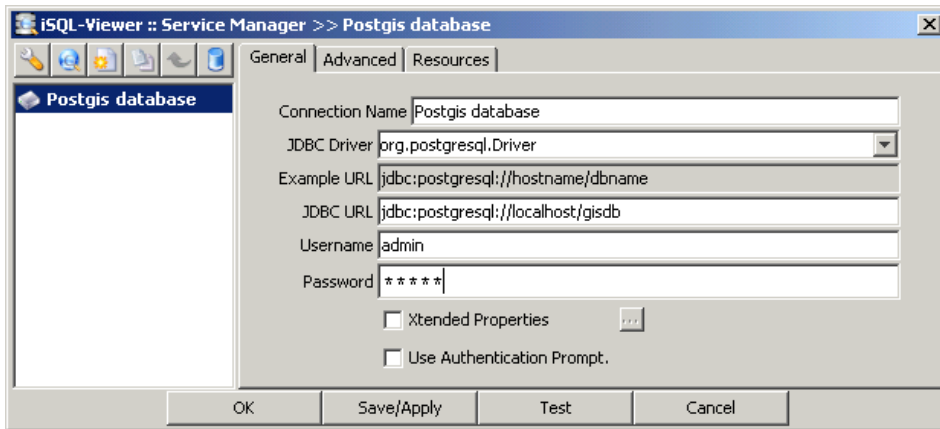


Figure 56

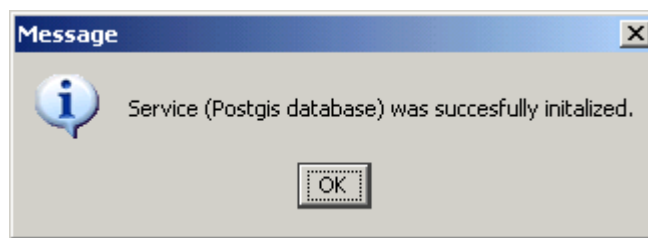


Figure 57

Test the tool by using a simple query (figure 68).

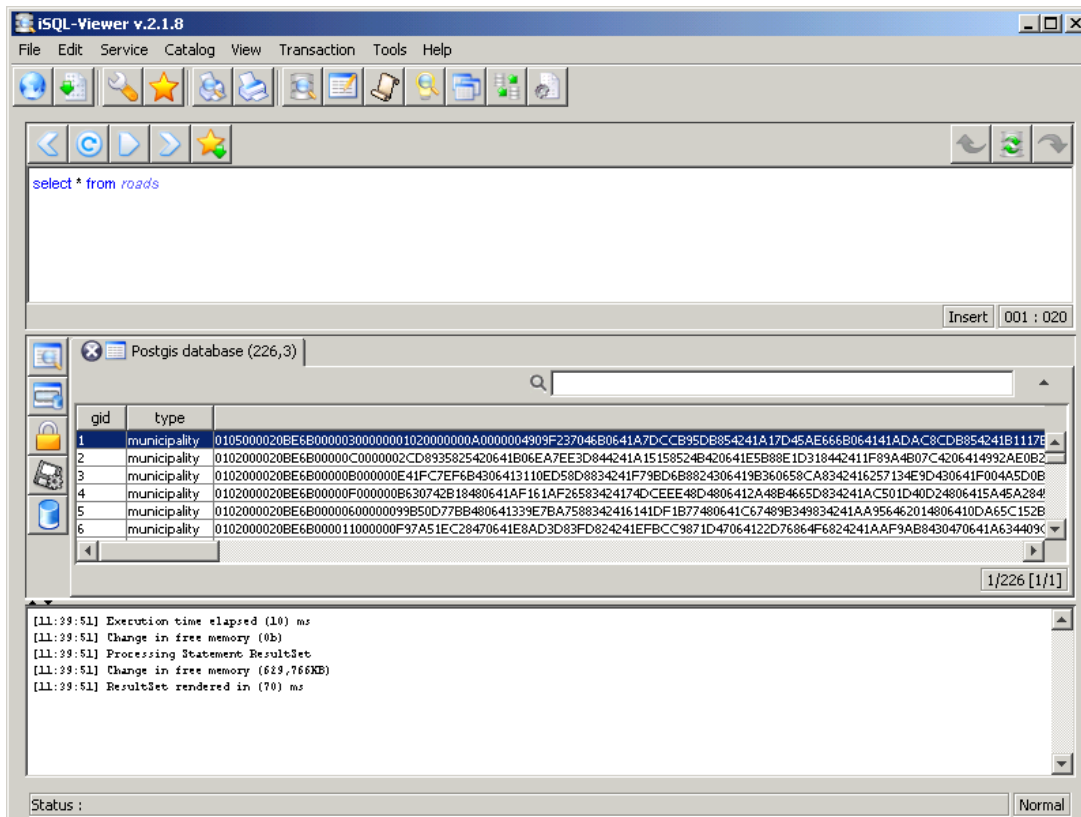


Figure 58

### 1.2.2.2 Practises

Find parcels contains in wsmunicipality named QUEMPERVEN (figure 59)

```
SELECT landcover2000.type FROM landcover2000, wsmunicipalities WHERE
contains(wsmunicipalities.the_geom, landcover2000.the_geom) AND
wsmunicipalities.name_mun='QUEMPERVEN'
```

To load the query result in OpenJUMP, you must use the same syntax like ExecuteQueryDialog plugin. The SQL statement would be :

```
SELECT asewkt(landcover2000 .the_geom) as the_geom, landcover2000.type FROM
landcover2000, wsmunicipalities WHERE contains(wsmunicipalities.the_geom,
landcover2000.the_geom) AND wsmunicipalities.name_mun='QUEMPERVEN'
```

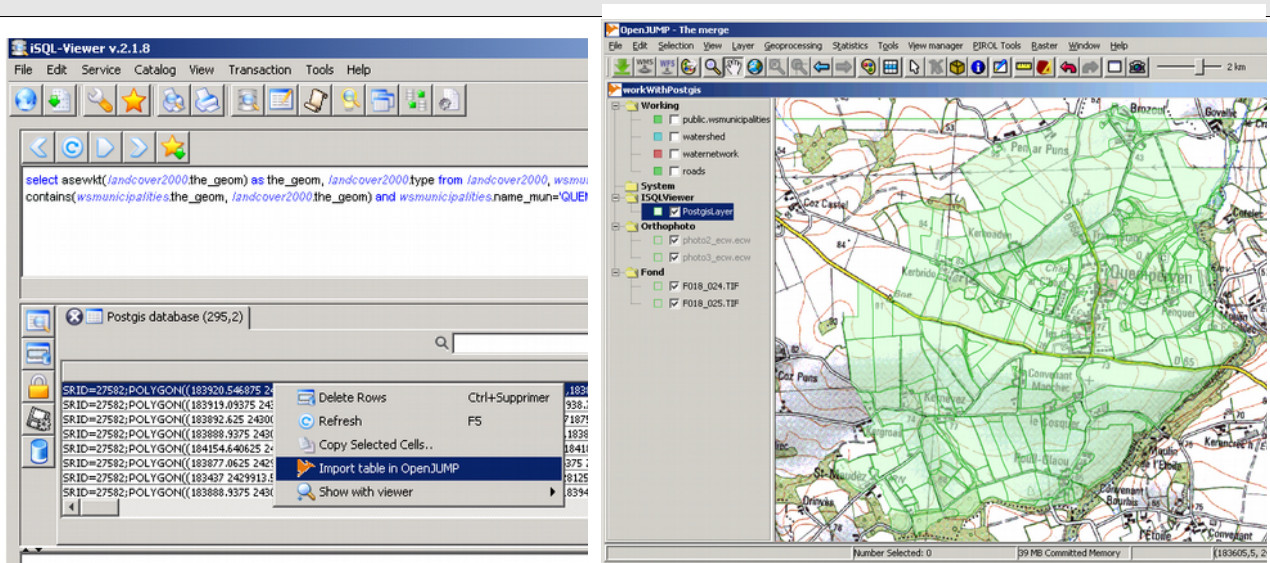


Figure 59

Create a buffer on rivers where the 20 distance value represents a high risk of pollution by pesticides. (figure 60)

```
SELECT asewkt(the_geom, 20)as
the_geom FROM rivers
```

Set new layer in red color and save it. Put the name buffrivers



Figure 60

Find parcels intersects the buffer layer

```
SELECT asewkt(landcover2000.the_geom)
as the_geom, landcover2000.type FROM
landcover2000, buffrivers WHERE
intersects(buffrivers.the_geom,
landcover2000.the_geom)
```

Set new layer in yellow color.



Figure 61

*All in one*

```
SELECT asewkt(landcover2000.the_geom) as the_geom, landcover2000.type FROM
landcover2000,rivers WHERE
intersects(buffer(rivers.the_geom,20),landcover2000.the_geom) AND rivers.the_geom &&
landcover2000.the_geom
```

To use this you will create a spatial index for each tables. The SQL statement is :

```
CREATE INDEX landcover2000_gidx on landcover2000 using gist ( the_geom
gist_geometry_ops );
VACCUANALYZE;
```

Unfortunately, there is a bug in iSQLViewer with **PostGIS** operators && , & < , and & > . However, you can test it with the simple postgis plugin.

Store complex statement in a view

For example, if you want to know how hedges touched a parcel with contain cereals. You need to use often this.

```
CREATE VIEW hedgerow.the_geom FROM hedgerow, landcover2000 WHERE
contains(landcover2000.the_geom, hedgerow.the_geom) AND landcover2000.type='cereals'
```

### 1.2.2.3 Watershed statistics : calculate and diffuse

The aim is to calculate environmental statistics by watersheds. This statistics will be published on internet in a html page and inform the decision maker about a potential pollution risk by pesticides.

#### Calculate statistics

What is the total length of the rivers in the big watershed and add the result in a column named riverlength

First: add a column in watershed layer.

```
ALTER TABLE watershed ADD COLUMN riverlength float8;
```

Secondly: populate the column

```
UPDATE watershed SET riverlength=sum(length(rivers.the_geom)) FROM rivers;
```

Lastly : display result

```
SELECT * FROM watershed;
```

In iSQLViewer you can execute all statements only once (figure 62).

To delete a column : ALTER TABLE DROP COLUMN

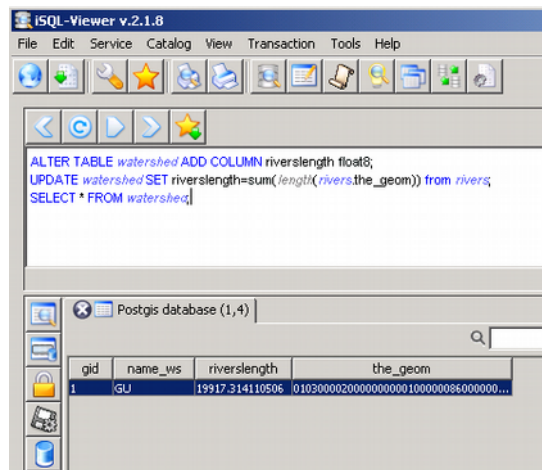


Figure 62

What is the density in meters by hectare of all hedges contains in the big watershed. Add the result in a column named hedgesdens

Firstly: add three columns in watershed layer.

```
ALTER TABLE watershed ADD COLUMN wsarea float8;  
ALTER TABLE watershed ADD COLUMN hedgeslength float8;  
ALTER TABLE watershed ADD COLUMN hedgesdens float8;
```

Secondly: calculate wsarea

```
UPDATE watershed SET wsarea=(area(watershed.the_geom)/10000);
```

Thirdly: create a view

```
CREATE VIEW hegdebigws as SELECT  
intersection(watershed.the_geom,hedgerow.the_geom) as the_geom FROM watershed,  
hedgerow;
```

Fourthly : load result in OpenJUMP

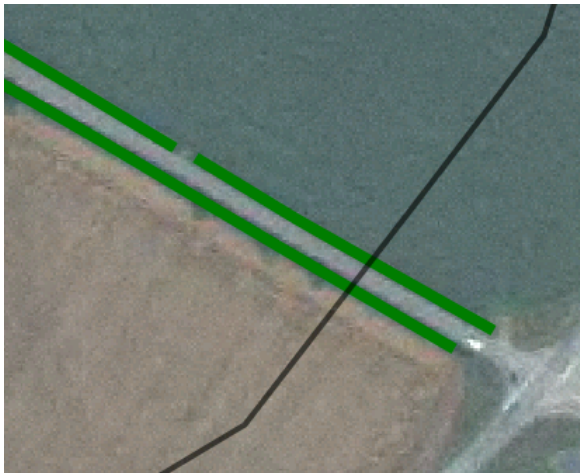


Figure 63 : Hedgerow layer

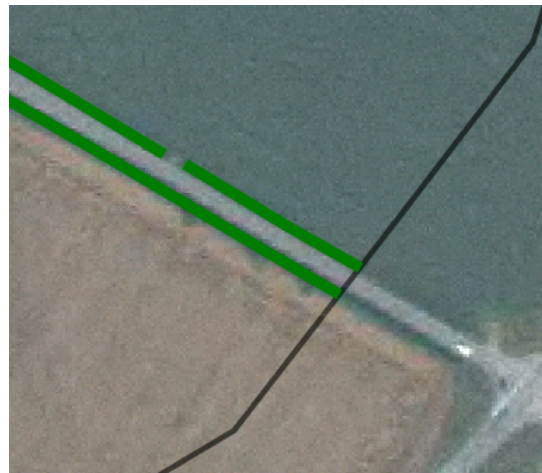


Figure 64 : View hegdebigws

Fifthly : set up the column hedgeslength

```
UPDATE watershed SET hedgeslength=(sum(length(hegdebigws.the_geom))) FROM  
hegdebigws;
```

Sixthly : calculate density

```
UPDATE watershed SET hedgesdens=hedgeslength/wsarea;
```

Lastly : display result and cartography it

```
SELECT * FROM watershed;
```

## Export watershed map and statistics in a simple html file

Firstly : Export watershed statistics in a html page.

Run this query (figure 65)

```
SELECT name_ws, riverslength,
wsarea,hedgeslength,
hedgesdens FROM watershed
```

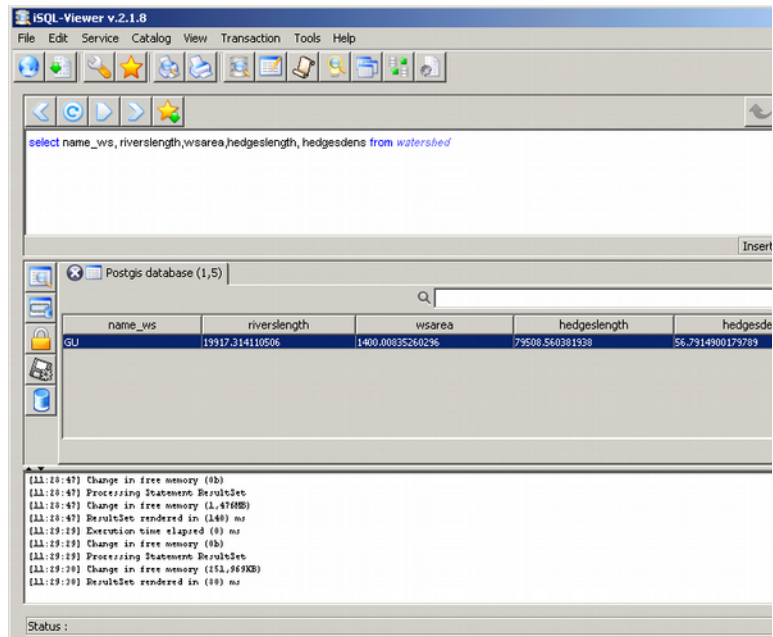


Figure 65

Turn on File>Export>Export data (figure 66)

Select HTML format and set the parameters like picture

Click next>next and save your page with the name mysimplepage.html.

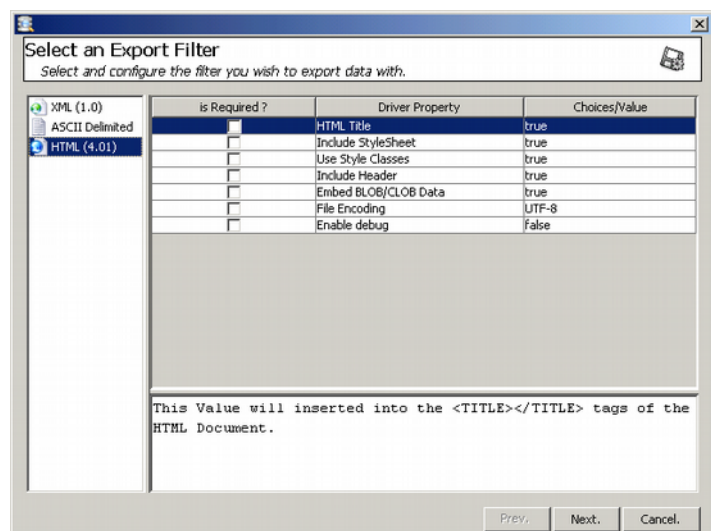



Figure 66

Create a map with watershed, hedgerow and rives.

Symbolise watershed  
ColorTheme based on name\_ws  
attribute

Select preset style 

Set label

Watershed

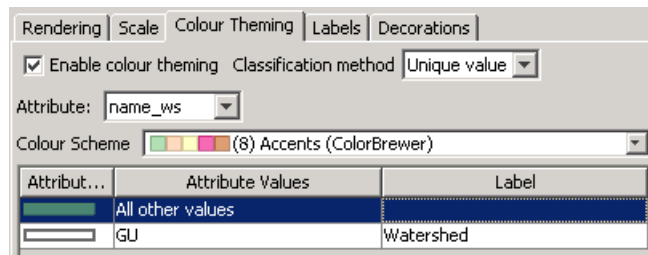


Figure 67

Symbolise hedgerow  
ColorTheme based on type attribute

For hedgeTalus

Unchek fill

Set line pattern = 7

Set green lines width=3

Change label like Hedge on talus

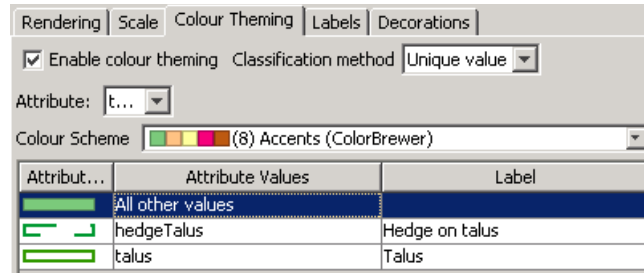


Figure 68

For talus

Unchek fill

Set green lines width=3

Change label like Talus

Symbolise rivers

ColorTheme based on type attribute

Unchek fill

Set blue lines width=3

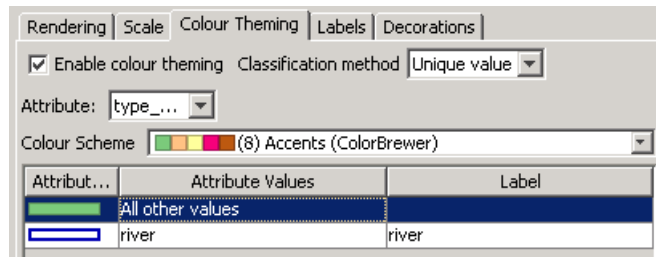


Figure 69

In view zoom to watershed and add scale bar (figure 70).

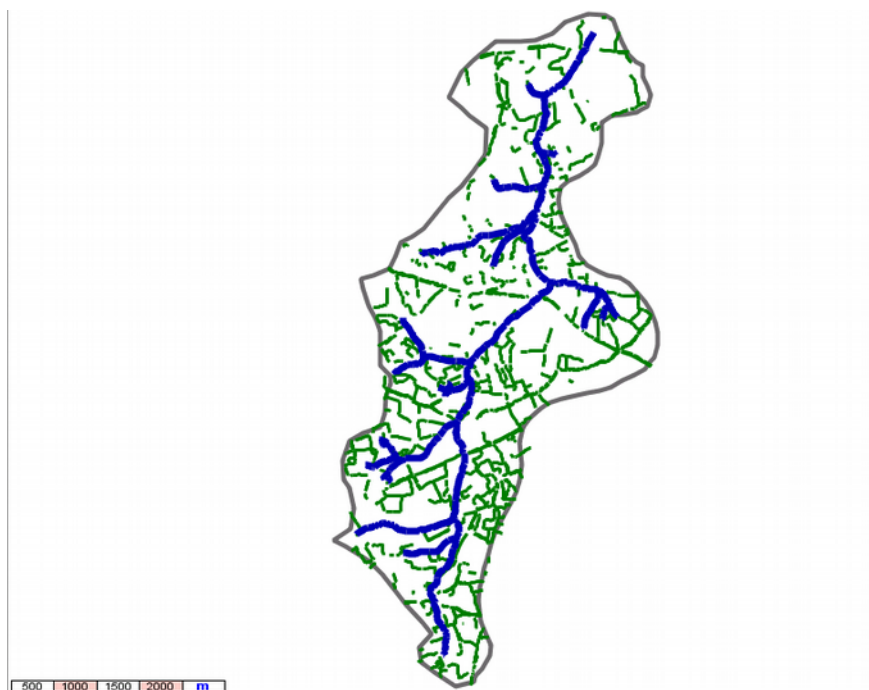


Figure 70



Now export each legends is the same folder like mysimplepage.html (figure 71).

Select layer, right click > Save legend as png.

Names :

hedglegend.png

riverslegend.png

watershedlegend.png

Integrate map in html page

Export view in svg. Turn on File > Save image in SVG format  
Name map.svg

Open map.svg in a text editor.  
Note width and height value

Open map.svg with inkscape  
Add legends and save

Open mysimplepage.html in a text editor. Below body add

Run mysimplepage.html in your web browser

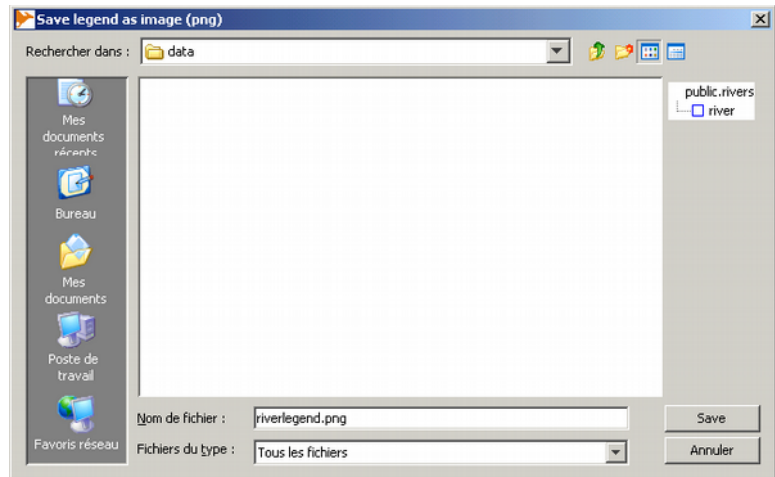


Figure 71

```
<rect x="0" y="0" width="813" style="stroke:none;" height="621" />
```

```
<body>
<embed src="map.svg" width="813" height="621"
type="image/svg+xml" name="emap">
</div>
```

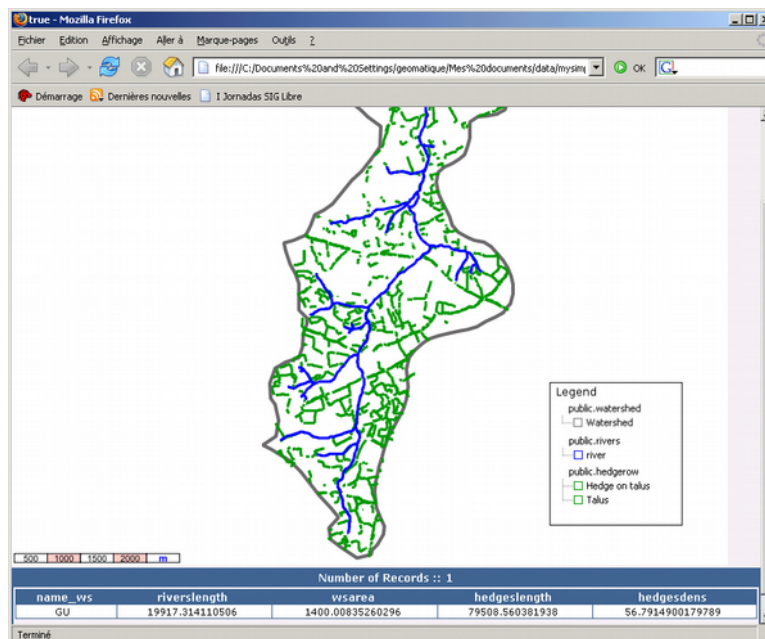


Figure 72

## Exercise

Calculate statistics by watersheds :

hedgerow density  
road density by type  
rivers length

Display result in html page with a map witch represents parcels risk watershed, subwatersheds, roads and rivers

## Notes

Merge roads by type

```
SELECT geomunion(the_geom),type FROM roads GROUP BY type;
```

Display geometry type

```
SELECT gid,geometrytype(the_geom) FROM rivers;
```

Working with SRID

Get SRID

```
SELECT getsrid(the_geom) FROM landcover2000 LIKE 1;
```

Change SRID

```
SELECT updategeometrysrid(rivers, the_geom, <new srid value>);
```

Then, use setSRID to change geometries internal SRID:

```
SELECT setsrid(the_geom, <new srid value>) FROM rivers;
```