

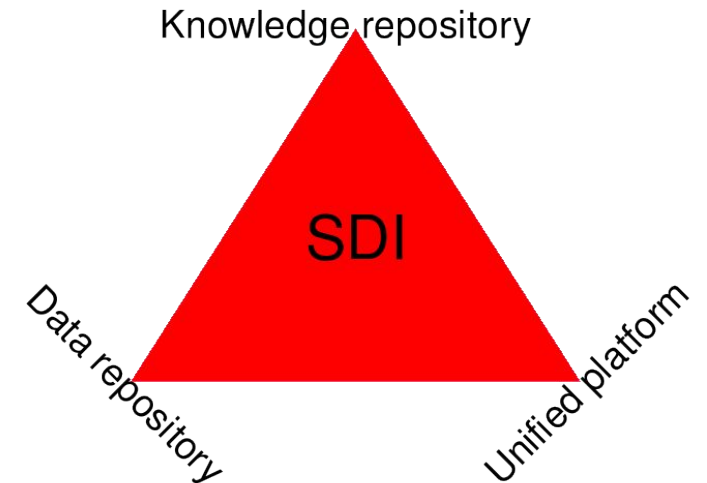
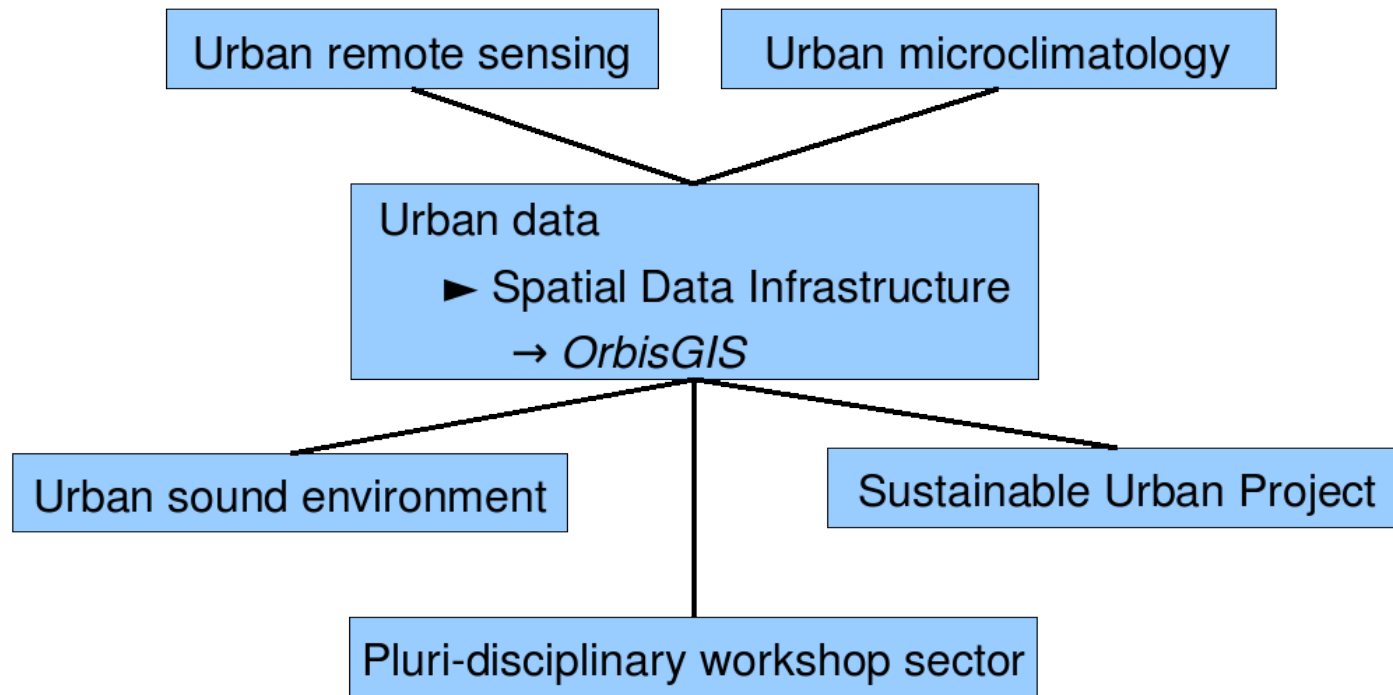
GDMS-R: a mixed SQL to manage raster and vector data

T. Leduc, E. Bocher, F. González Cortés, and G. Moreau
IRSTV – CERMA Laboratory – Ecole Centrale de Nantes
GIS 2009, Ostrava (Czech Republic), January 26th 2009



- Context and motivations,
- Internal architecture and data model,
- Some use cases,
- Need of some enhancement...

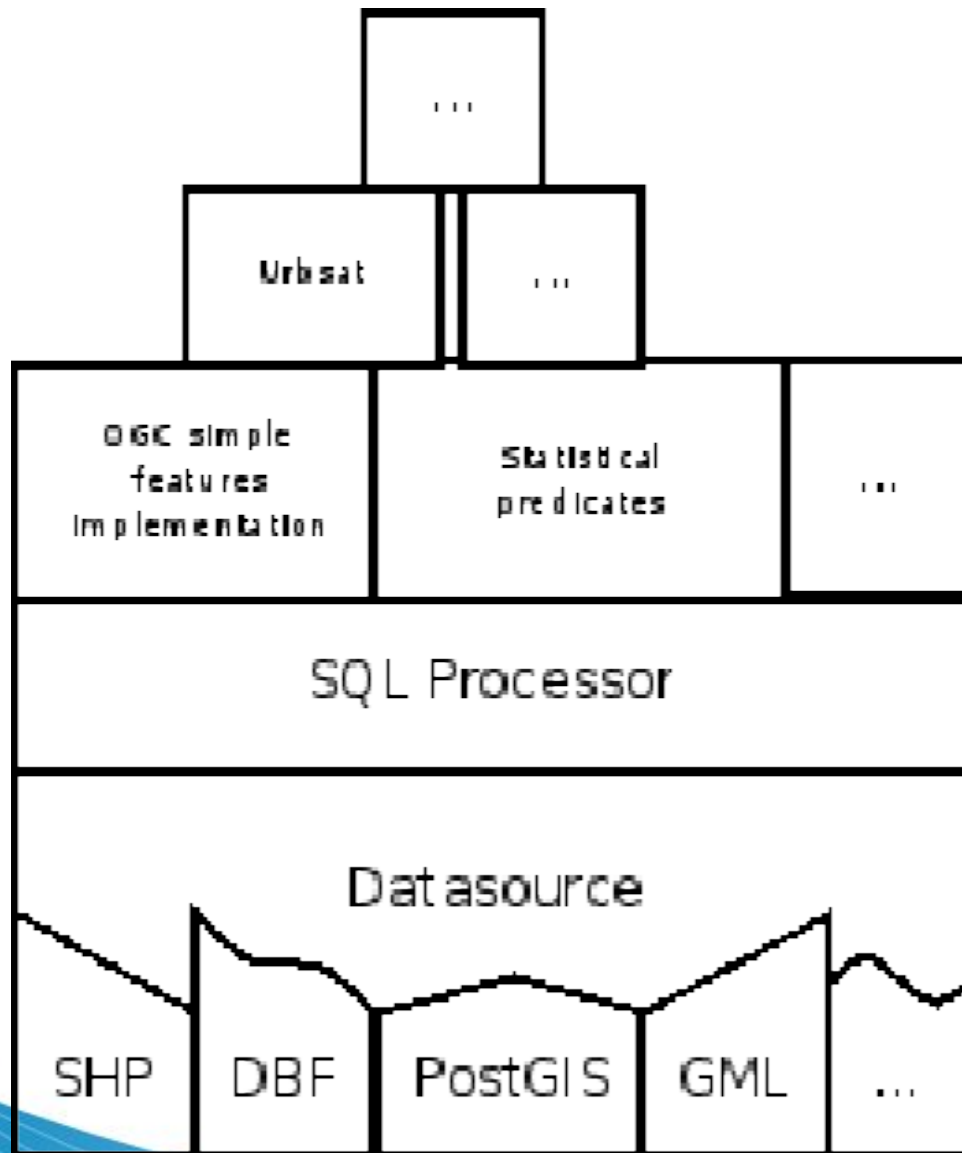
The context



- Deals with several formats (some of them are mandatory due to existing simulation software) but also with complex and heterogeneous simulation softwares,
- Respect users practices constraints but also international standards,
- Provides pre-processed data for the SDI,

⇒ GDMS: Generic Data source Management System = API + extensible OGC compliant SQL.

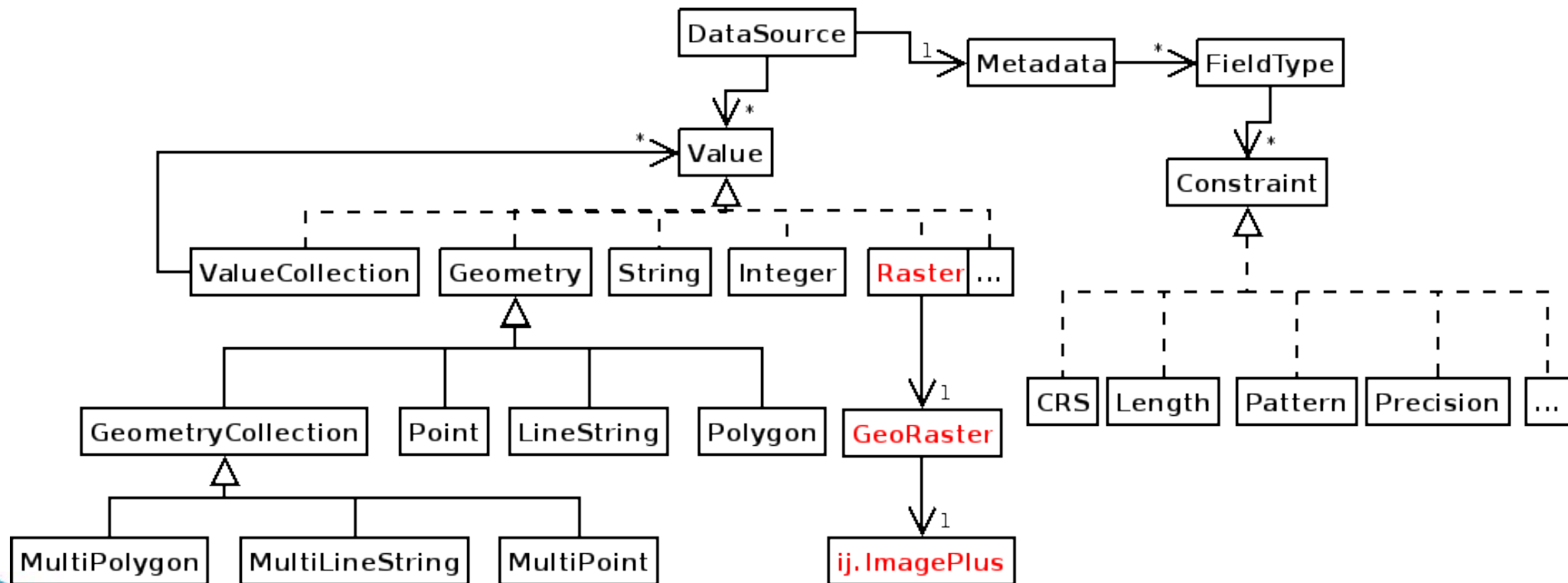
Internal architecture



- a layered architecture that stacks:
 - a driver layer,
 - an adapter layer,
 - an application layer...
- that can be extended
 - I/O,
 - functionalities.

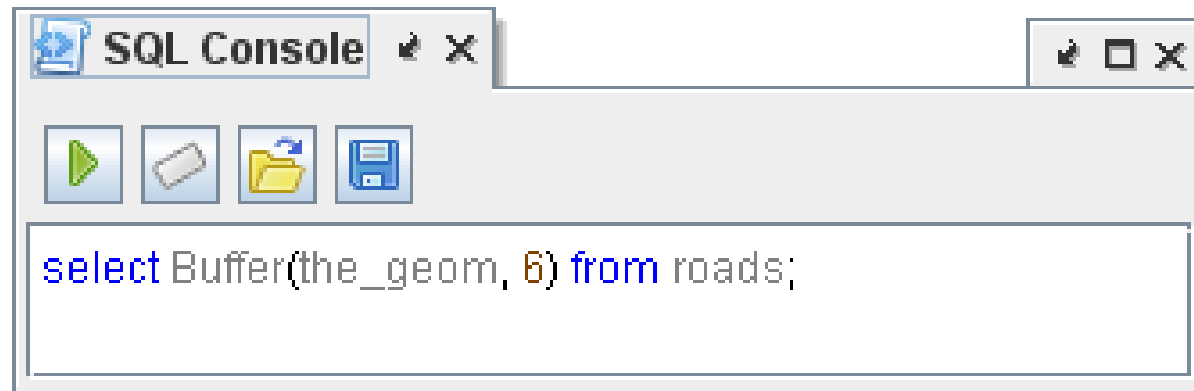
Extended data model

- <DataSource, row, field> close the GeoAPI model <FeatureCollection, Feature, Feature attribute>. A typical tabular model close to the JDBC standard.



GDMS in action

- simple semantic use case (1):



before...

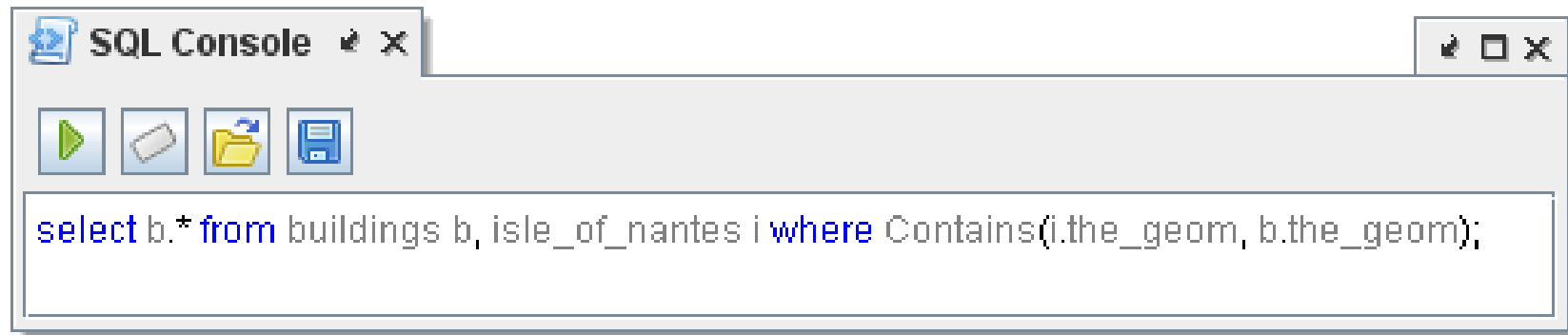


... and after



GDMS in action

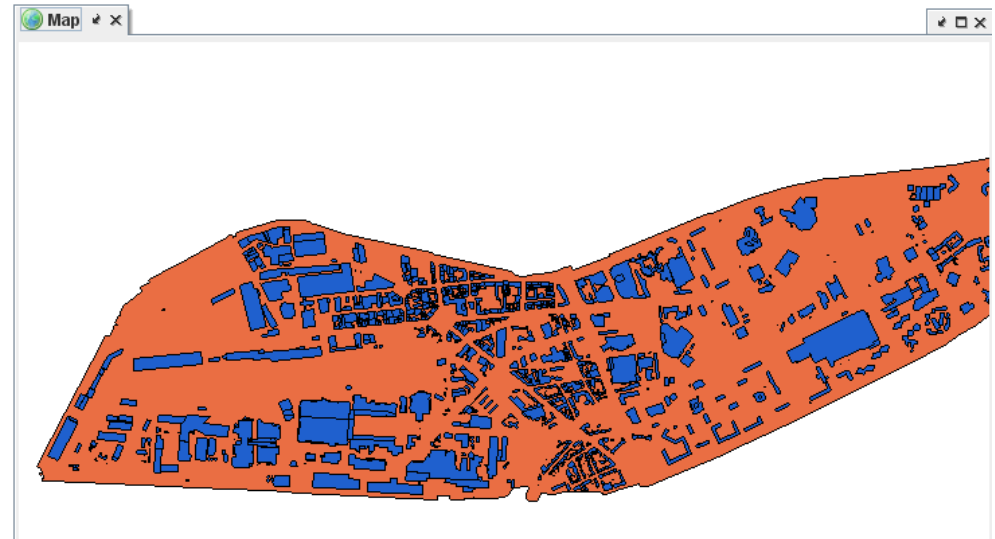
- simple semantic use case (2):



before...

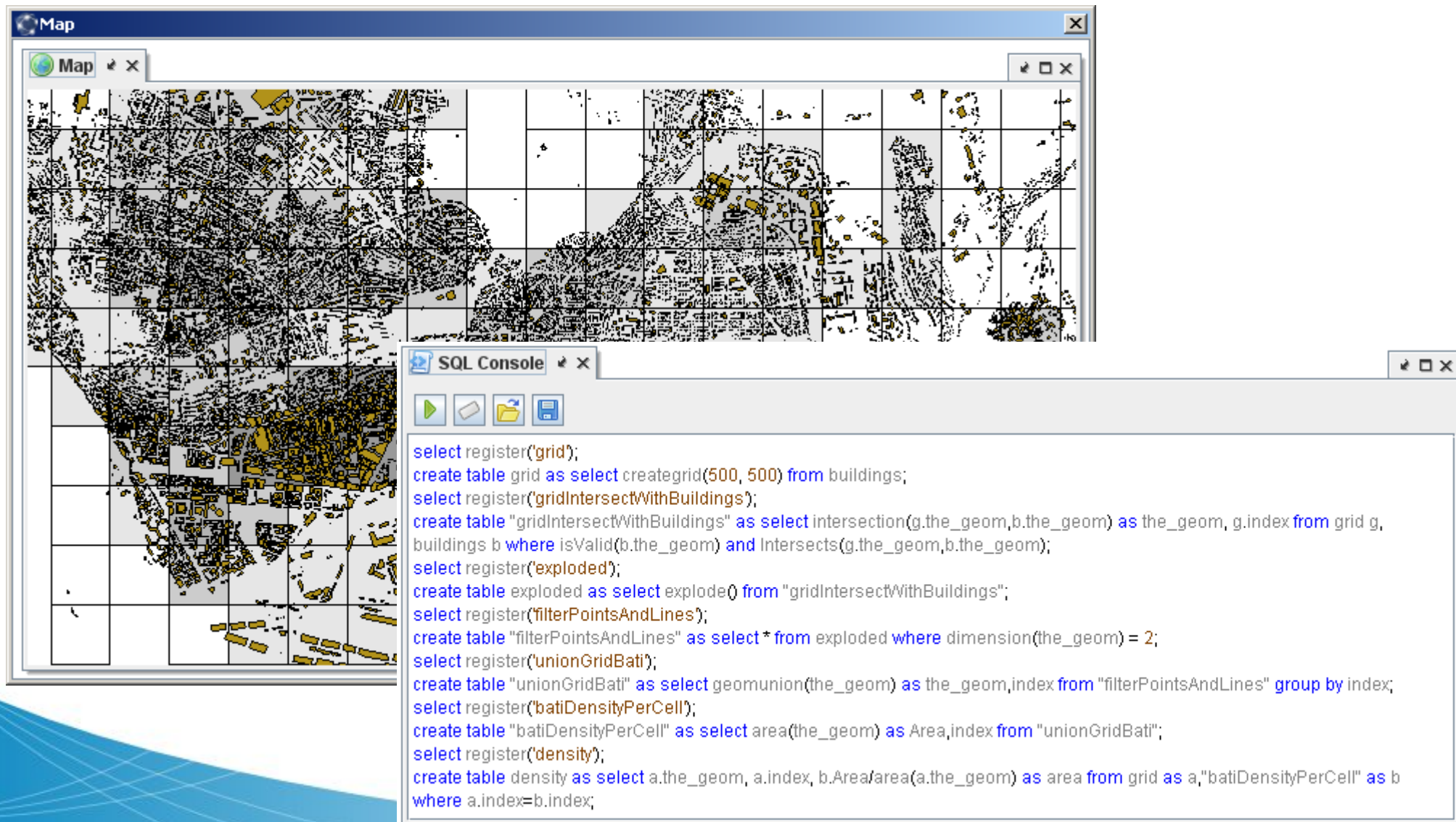


... and after



GDMS in action

- urban indicators, UrbSAT plugin (1):

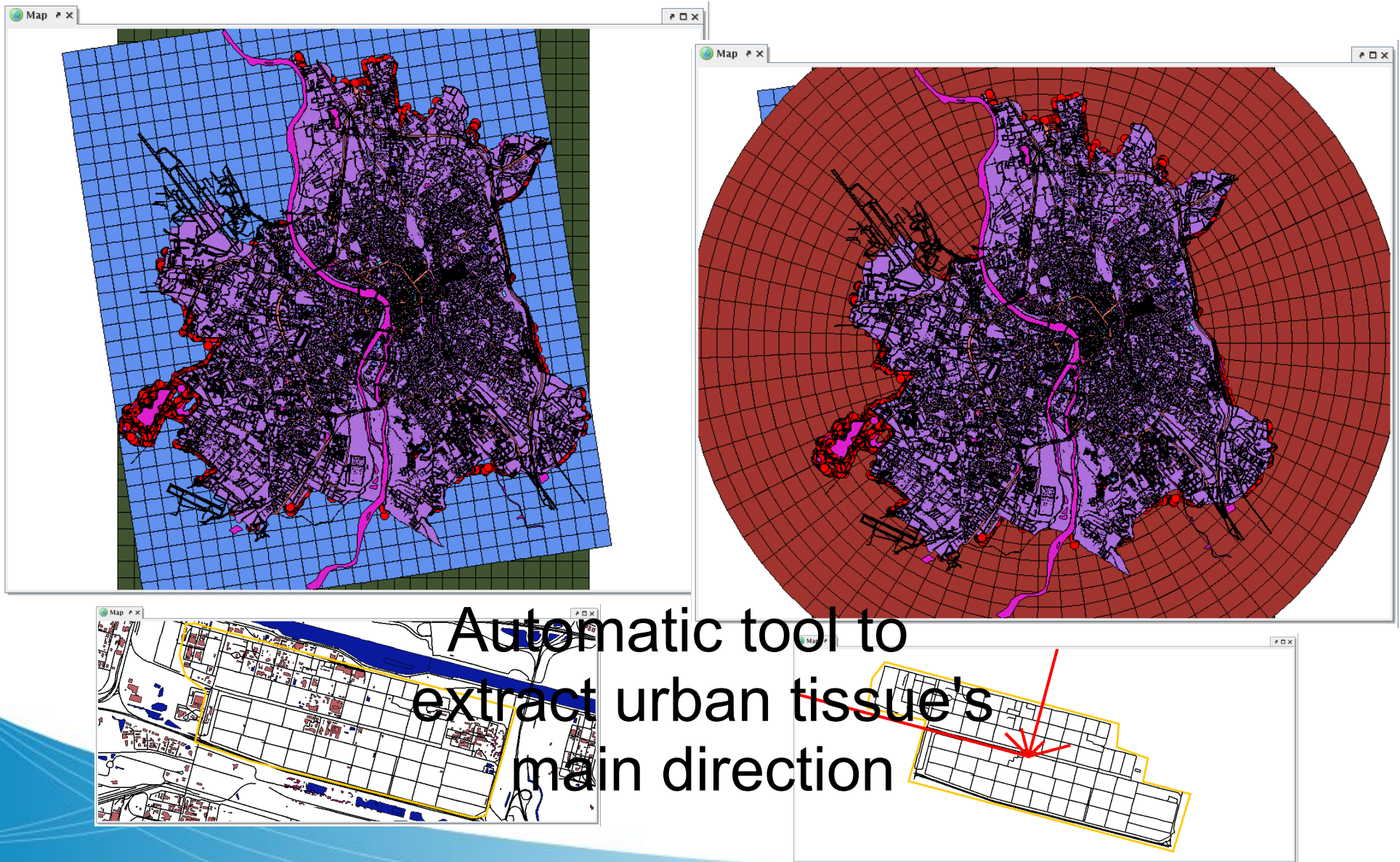


The screenshot displays the OrbisGIS interface. The 'Map' window shows a map of Europe with a grid overlay. The 'SQL Console' window is open, showing a series of SQL commands for creating and populating tables related to urban indicators.

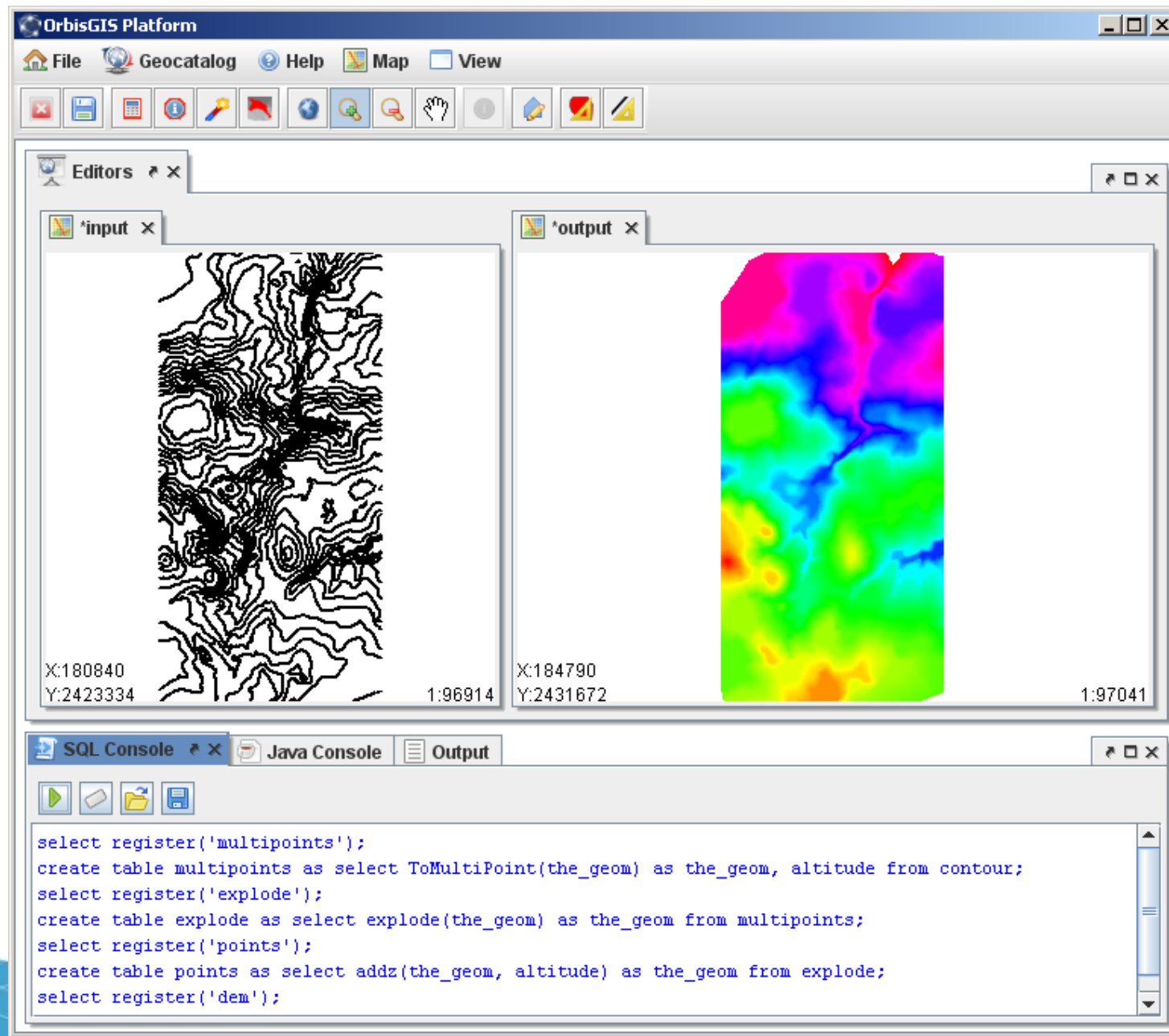
```
select register('grid');
create table grid as select creategrid(500, 500) from buildings;
select register('gridIntersectWithBuildings');
create table "gridIntersectWithBuildings" as select intersection(g.the_geom,b.the_geom) as the_geom, g.index from grid g,
buildings b where isValid(b.the_geom) and Intersects(g.the_geom,b.the_geom);
select register('exploded');
create table exploded as select explode() from "gridIntersectWithBuildings";
select register('filterPointsAndLines');
create table "filterPointsAndLines" as select * from exploded where dimension(the_geom) = 2;
select register('unionGridBati');
create table "unionGridBati" as select geomunion(the_geom) as the_geom,index from "filterPointsAndLines" group by index;
select register('batiDensityPerCell');
create table "batiDensityPerCell" as select area(the_geom) as Area,index from "unionGridBati";
select register('density');
create table density as select a.the_geom, a.index, b.Area/area(a.the_geom) as area from grid as a,"batiDensityPerCell" as b
where a.index=b.index;
```

GDMS in action

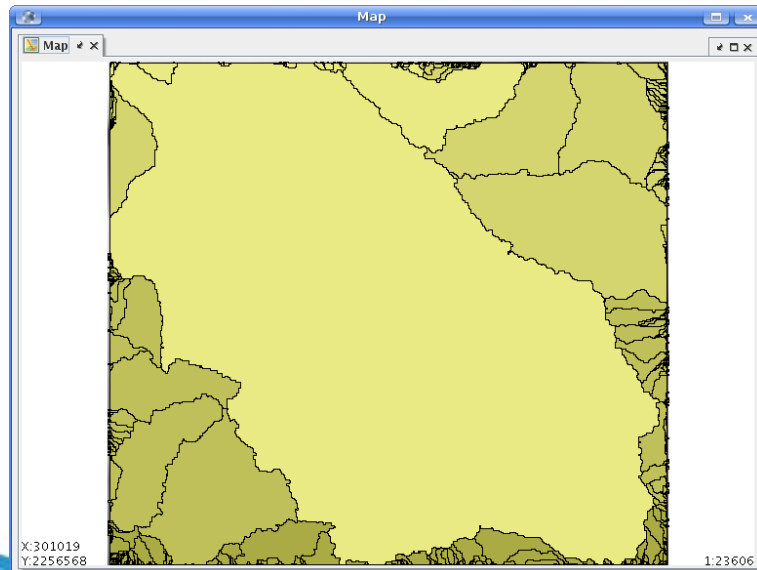
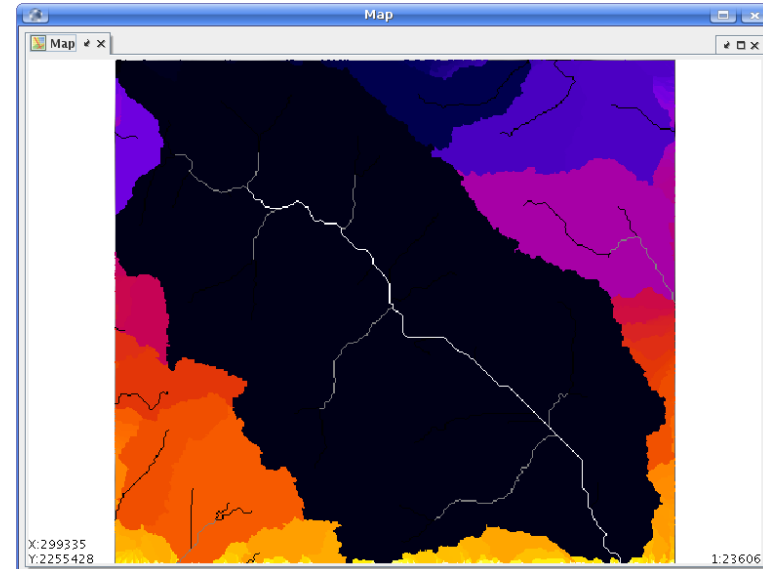
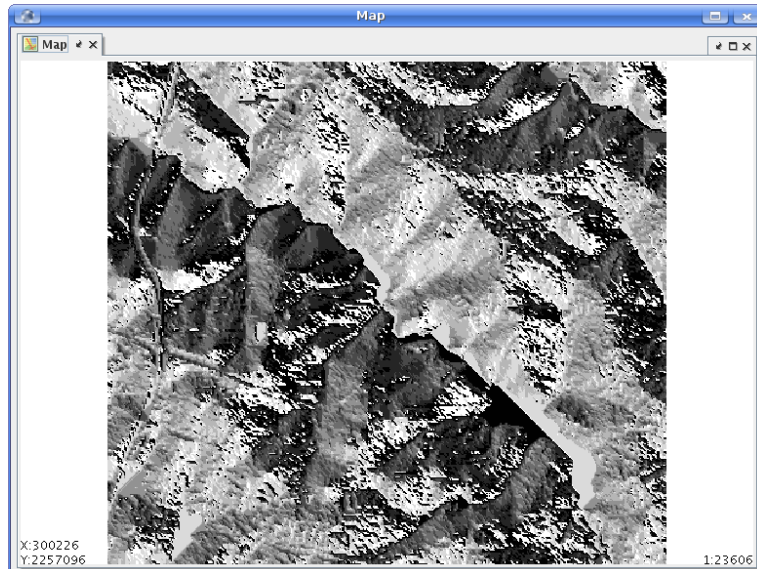
- urban indicators, UrbSAT plugin (2):



From vector to raster



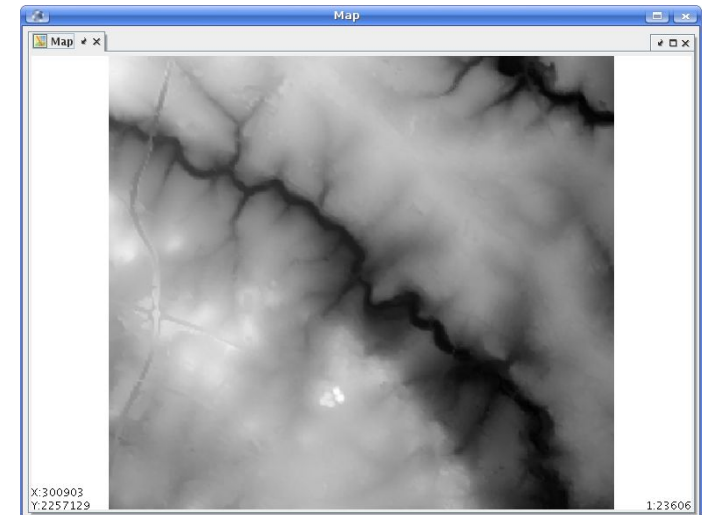
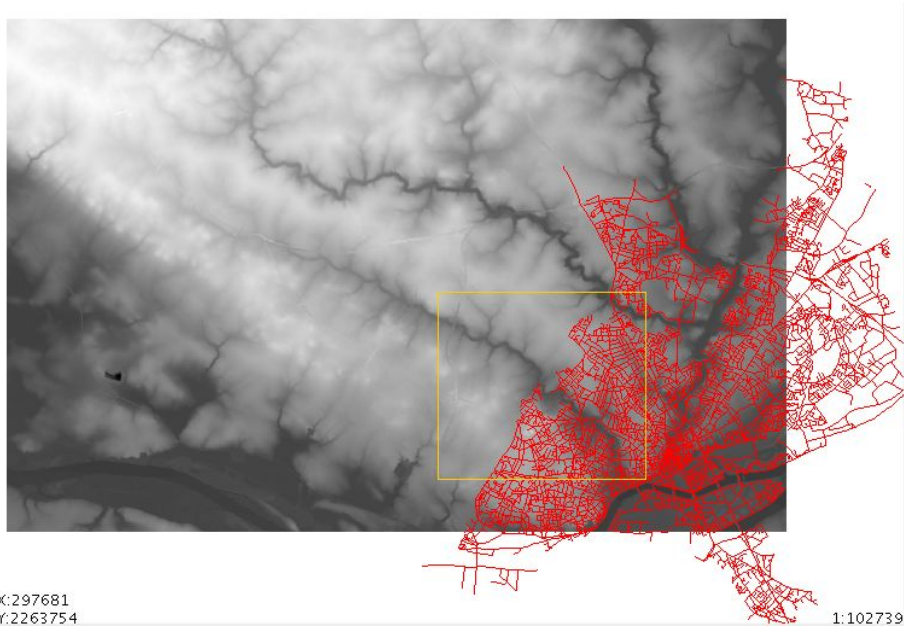
From raster to vector



```
SELECT  
  D8Watersheds(D8Direction(raster))  
from dem;
```

```
SELECT RasterToPolygons(raster)  
from watersheds;
```

Need of a mixed SQL



```
SELECT CropRaster(raster, the_geom) FROM dem, fence;
```

Road impact on runoff path



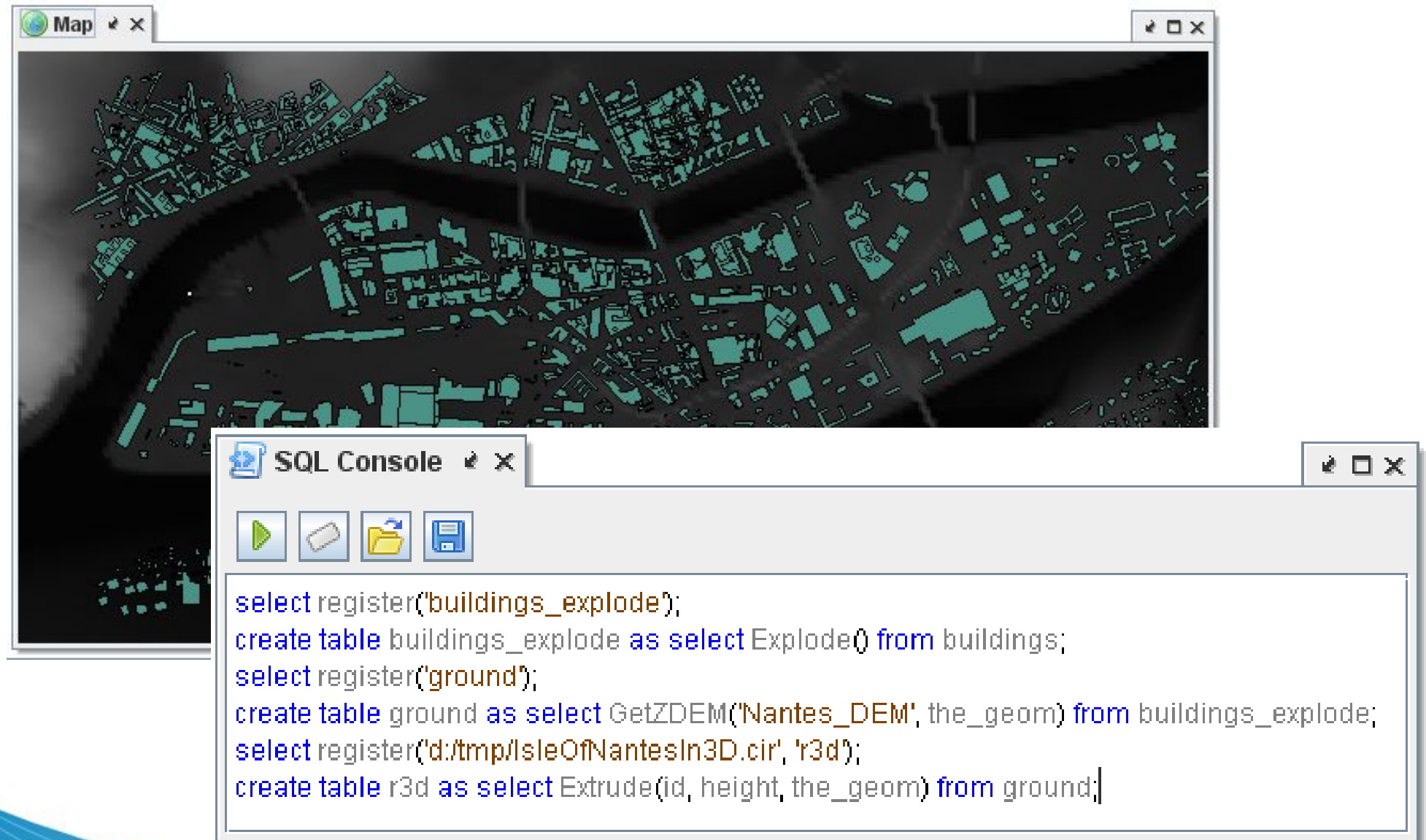
```
SELECT  
  D8Accumulation(D8Direction(raster)  
from dem;
```

```
SELECT RasterizeLine(the_geom,  
  raster,1) from roads, dem;
```

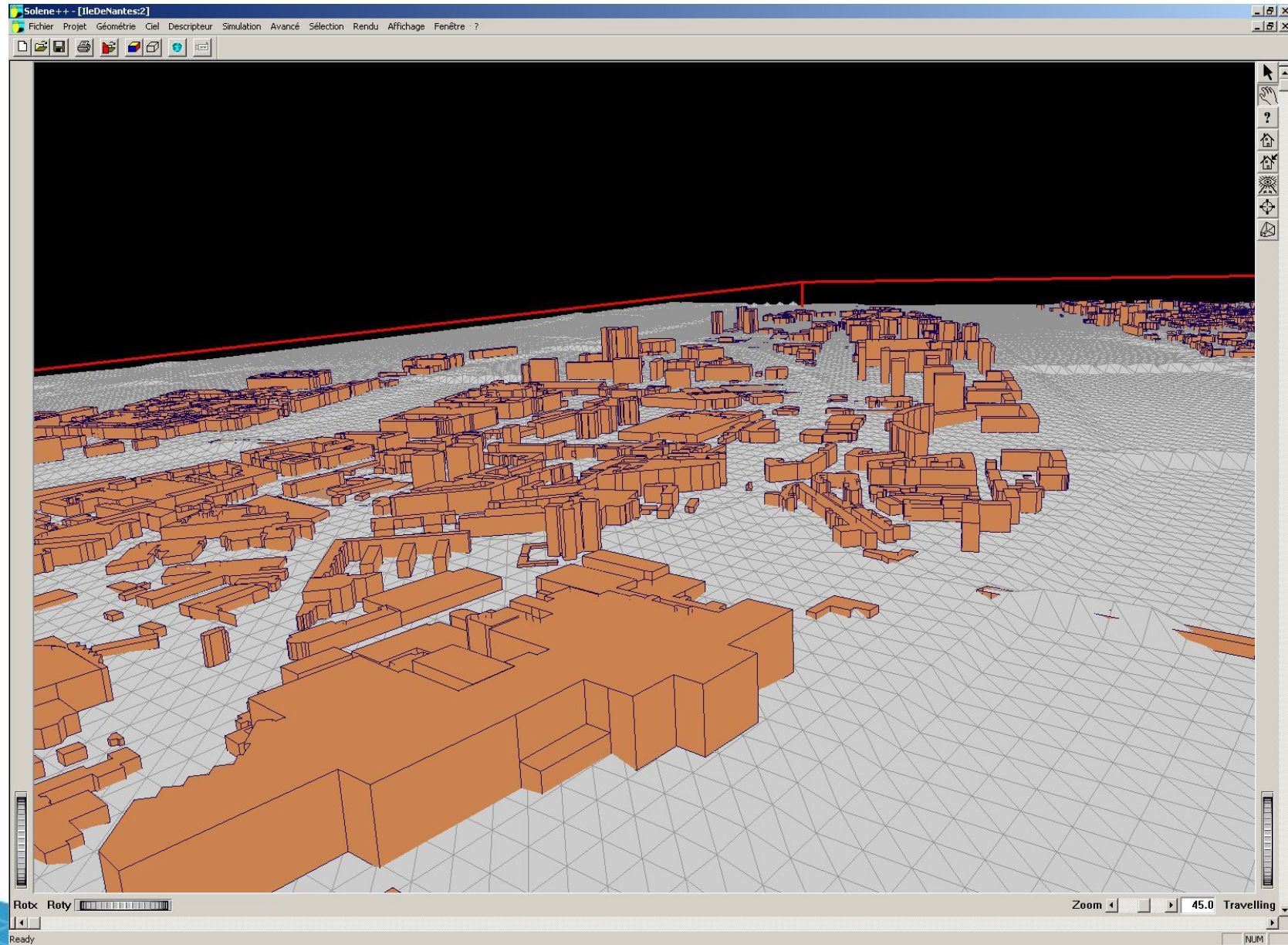
```
SELECT  
  D8ConstrainedAccumulation(d.raster,  
r.raster) from direction d,  
rasterized_roads r;
```



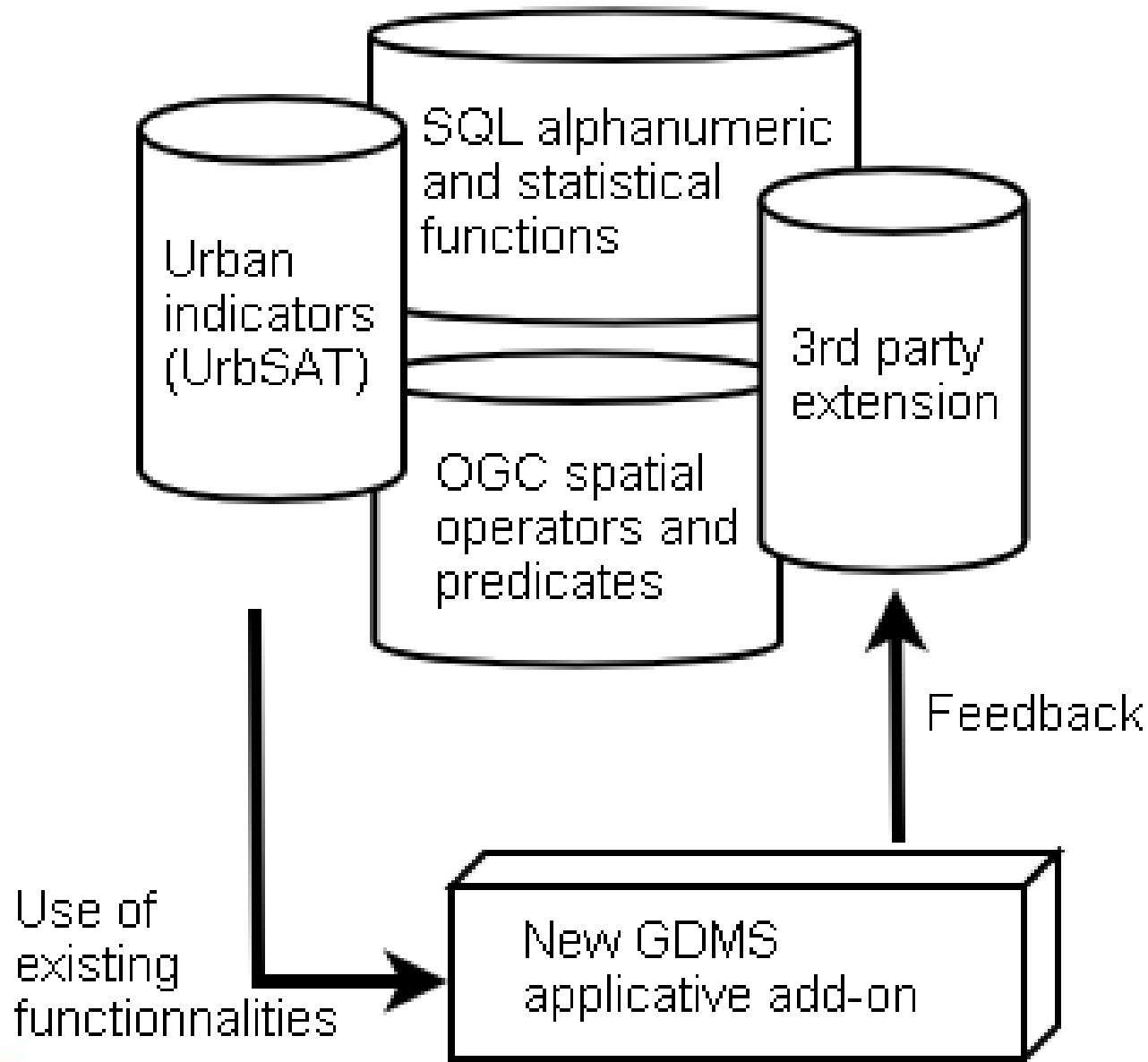
Mixing both raster & vector



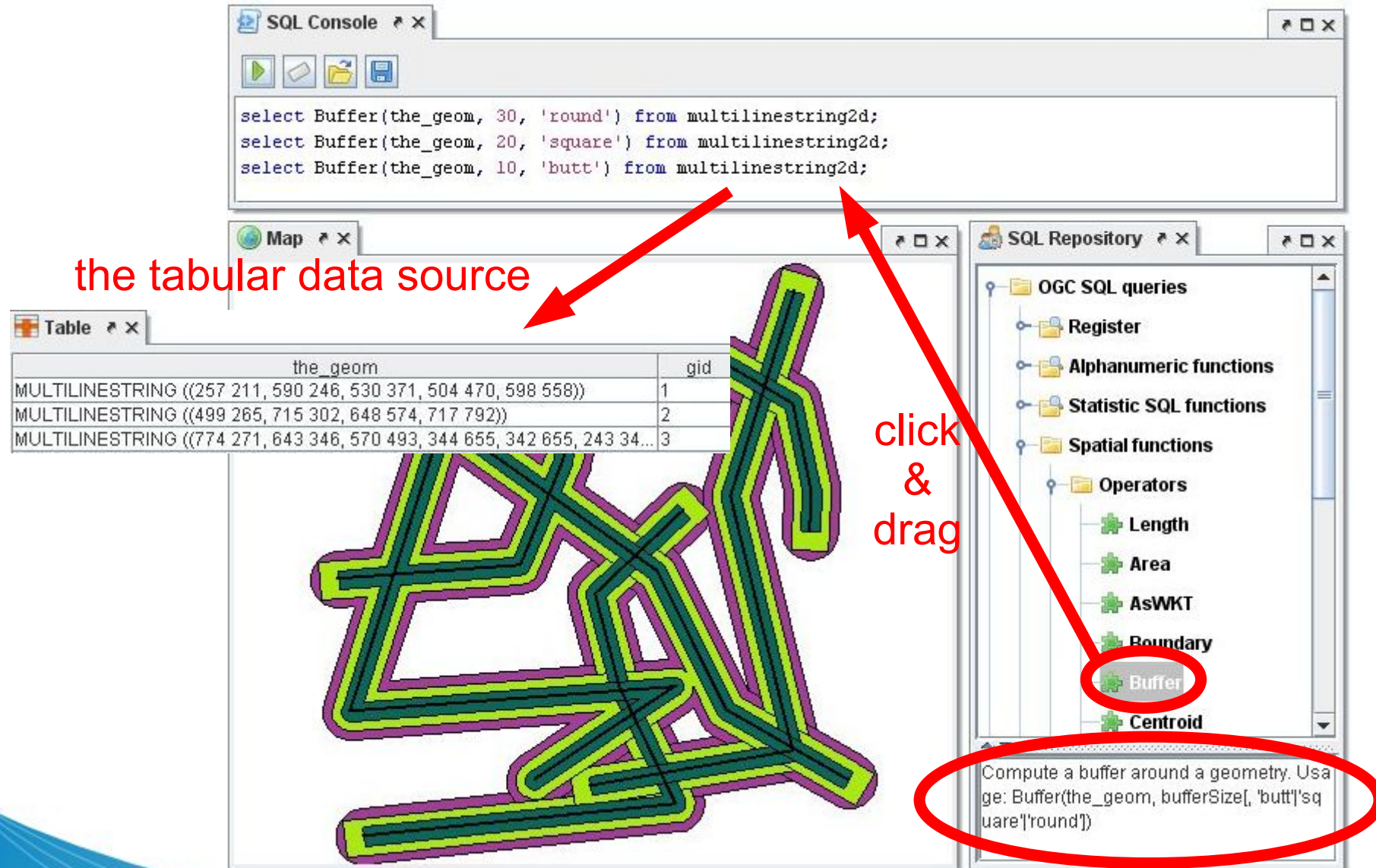
50,000 triangles (DEM) and
30,000 polygons (buildings)



GeoCognition concept (1)



GeoCognition concept (2)



the tabular data source

click & drag

the_geom	gid
MULTILINESTRING ((257 211, 590 246, 530 371, 504 470, 598 558))	1
MULTILINESTRING ((499 265, 715 302, 648 574, 717 792))	2
MULTILINESTRING ((774 271, 643 346, 570 493, 344 655, 342 655, 243 34...	3

SQL Console

```
select Buffer(the_geom, 30, 'round') from multilinestring2d;  
select Buffer(the_geom, 20, 'square') from multilinestring2d;  
select Buffer(the_geom, 10, 'butt') from multilinestring2d;
```

SQL Repository

- OGC SQL queries
 - Register
 - Alphanumeric functions
 - Statistic SQL functions
 - Spatial functions
 - Operators
 - Length
 - Area
 - AsWKT
 - Boundary
 - Buffer**
 - Centroid

Compute a buffer around a geometry. Usage: Buffer(the_geom, bufferSize, 'butt'|'square'|'round')

The SIF framework (API)



- a framework to build UI panels (forms) with an advanced input field validation mechanism,
- all interface's input fields are processed as if they were a single row in a SQL table,
- a SQL expression is built that returns a boolean result – it has to respect the where's clause !

```
DynamicUIPanel myDynamicUIPanel = UIFactory.getDynamicUIPanel(  
    "1st panel", null, new String[] { "field1", "field2" },  
    new int[] { SQLUIPanel.STRING, SQLUIPanel.DOUBLE },  
    new String[] { "field1 LIKE 'abc%'", "field2 > 12" },  
    new String[] { "field1 must start with 'abc'", "field2 must be greater than 12 !" });  
UIFactory.showDialog(myDynamicUIPanel);
```

Schemas and metadata (1)

- GDMS SQL syntax extension to provide SQL script re-usability without any modification,
- Provide a mechanism to obtain informations about internal operations so that 2 users interfaces can be used:
 - a wizard that ask for each of the required items the script uses,
 - a graphic modeler (graphic query builder) that lets the graphic connection between script inputs and outputs,
- Define and process schemas to filter tables based on their metadata.

Schemas and metadata (2)

-- Creates a buffer on the specified table

-- with the specified size and shape

```
DOC('Table containing the geometric field to create the buffer');
```

```
DECLARE source:TABLE(the_geom:GEOMETRY);
```

```
DOC('Buffer size. It must be positive');
```

```
DECLARE size:INT(value > 0);
```

```
DOC('Buffer type');
```

```
DECLARE buffer_type:STRING VALUES ('CAP', 'BUTT', 'ROUND') DEFAULT 'CAP';
```

```
BEGIN;
```

```
CREATE TABLE result AS SELECT buffer(the_geom, size, buffer_type) from source;
```

```
COMMIT;
```

- query and display 3D spatial data sets,
- integrate the temporal dimension,
- deals with multiplexing as outputs,
- from tabular to hierarchical approach...

Why not try it by yourself...



- a FOSS scientific GIS platform developed by a French research institute:

<http://orbisgis.cerma.archi.fr/>

- ... and hosted by a French academic forge:

<http://sourcesup.cru.fr/projects/orbisgis>

