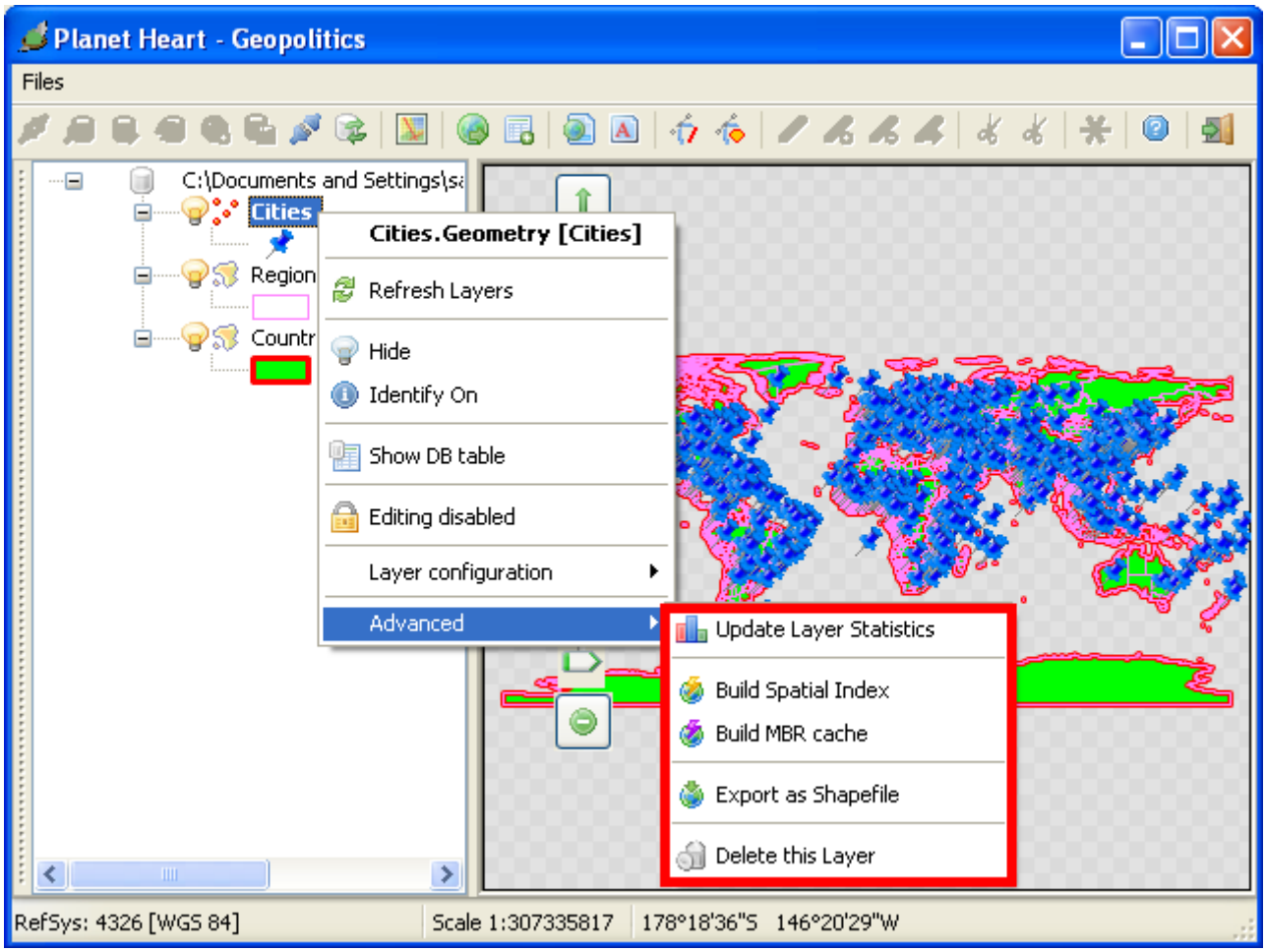
not at all a manual
simply a quick how-to-do guide

This section explains the **advanced** operations supported by **spatialite-gis**

We'll assume you've already acquired a sound familiarity with **basic** operations.

Table of contents:

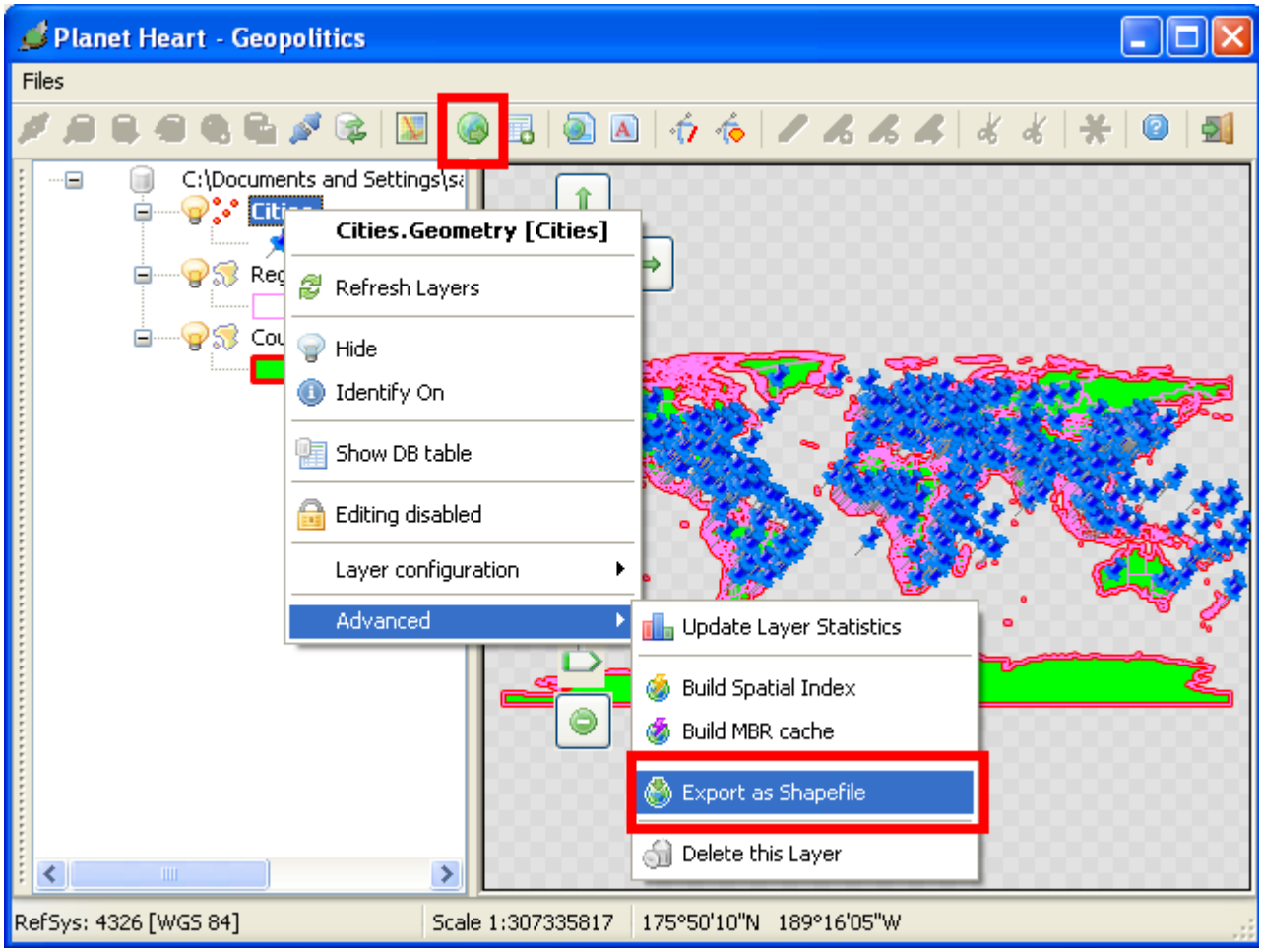
1- Layer management: extent / drop table / Spatial Index



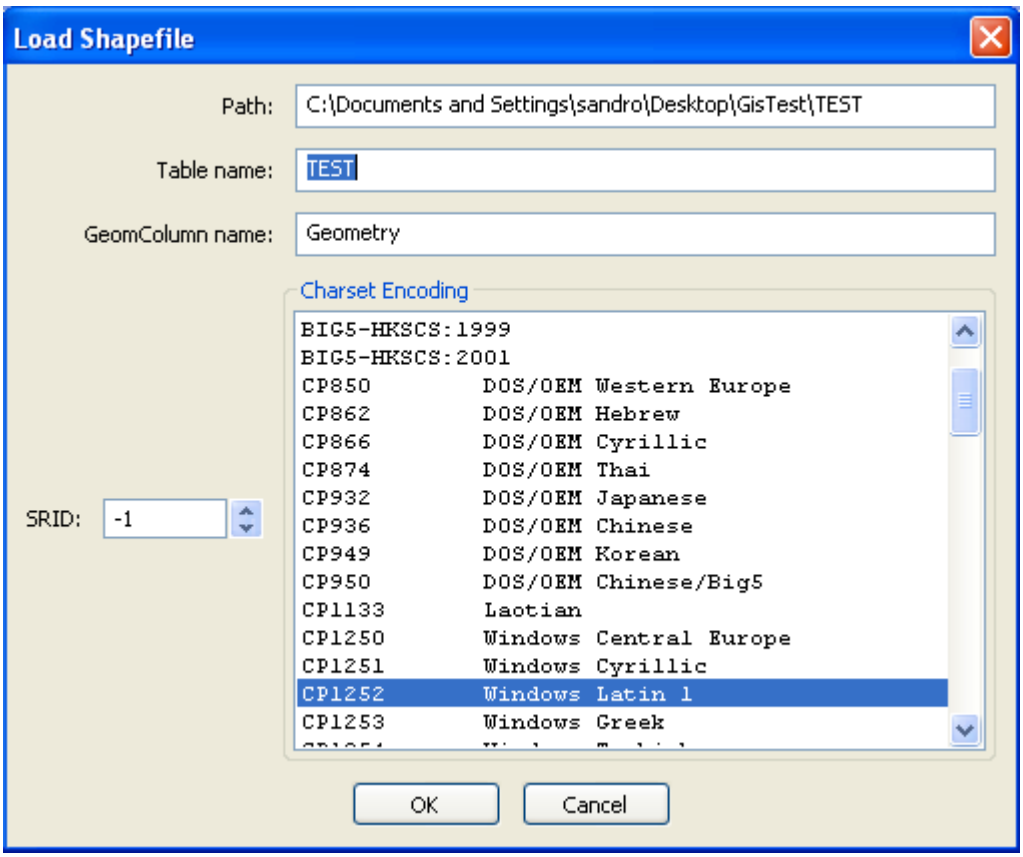
spatialite-gis allows you to perform various common utility task related with layer (*DB table*) management:

- you can **update the layer statistics**: this one is a batch, deferred task, specifically aimed to determine the **extent** for a single layer [*and the whole project's full extent*]. You are strongly suggested to update layer's statistics every time you've inserted, deleted or updated lots and lots of entities.
- you can create or remove a **Spatial Index** [either an **R*Tree** one, or an **MBR-cached** one, at your free choice]. Please note: creating a Spatial Index for a layer storing less than some thousands entities doesn't make any speed improvement at all. Conversely, creating a Spatial Index for a layer storing some hundred thousands [or millions] entities make a dramatic speed improvement.
- and finally you can **drop** [*delete*] a complete layer. In this occurrence spatialite-gis take care of dropping the corresponding DB table and any further layer-related definition. Please note: this one is an **unrecoverable operation**; so, think twice before confirm !!!

2 – Shapefile import / export

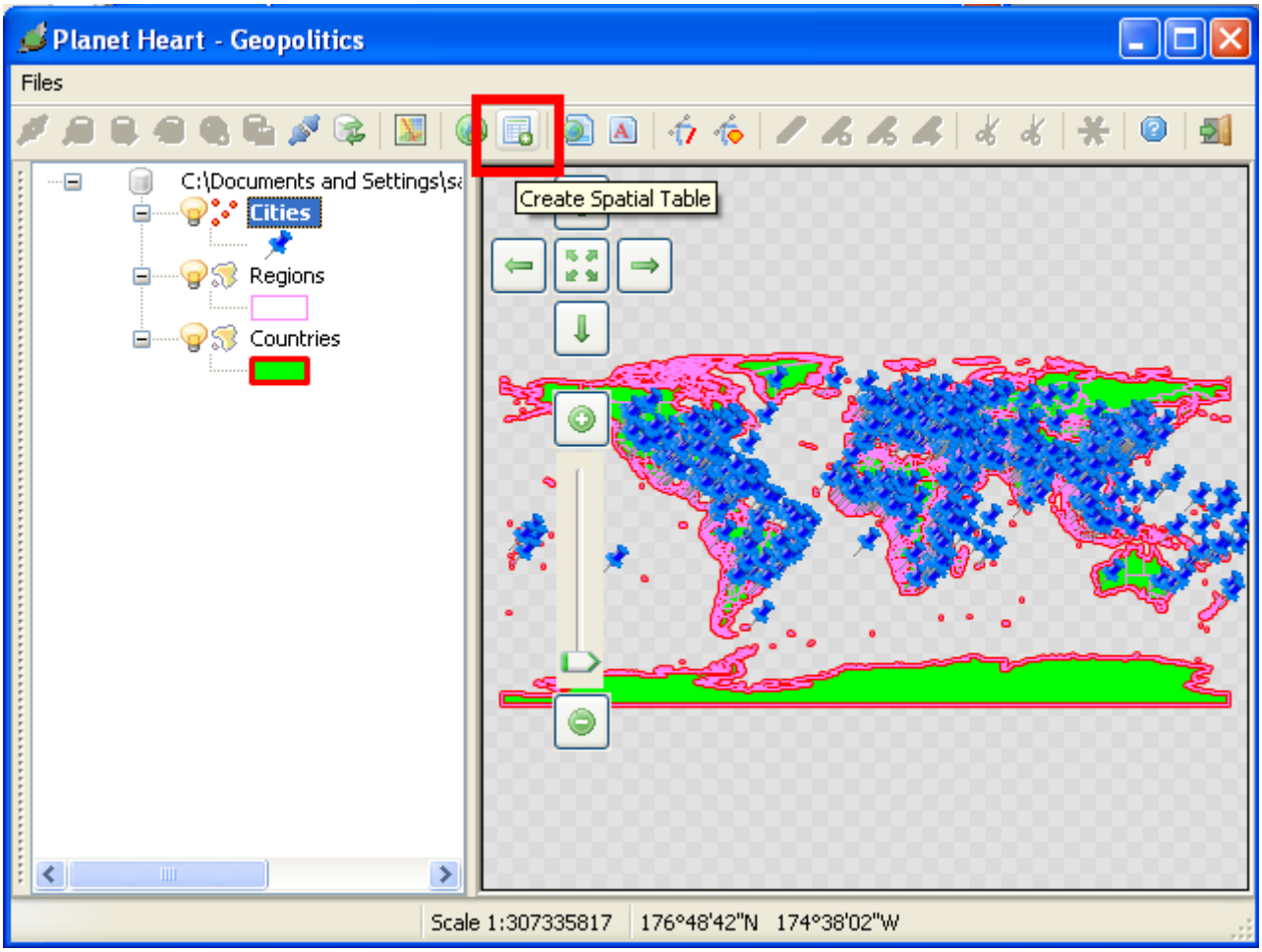


Importing and / or exporting GIS datasets using the popular and widespread **Shapefile** format usually is a common task: so **spatialite-gis** allow you to perform both operations.

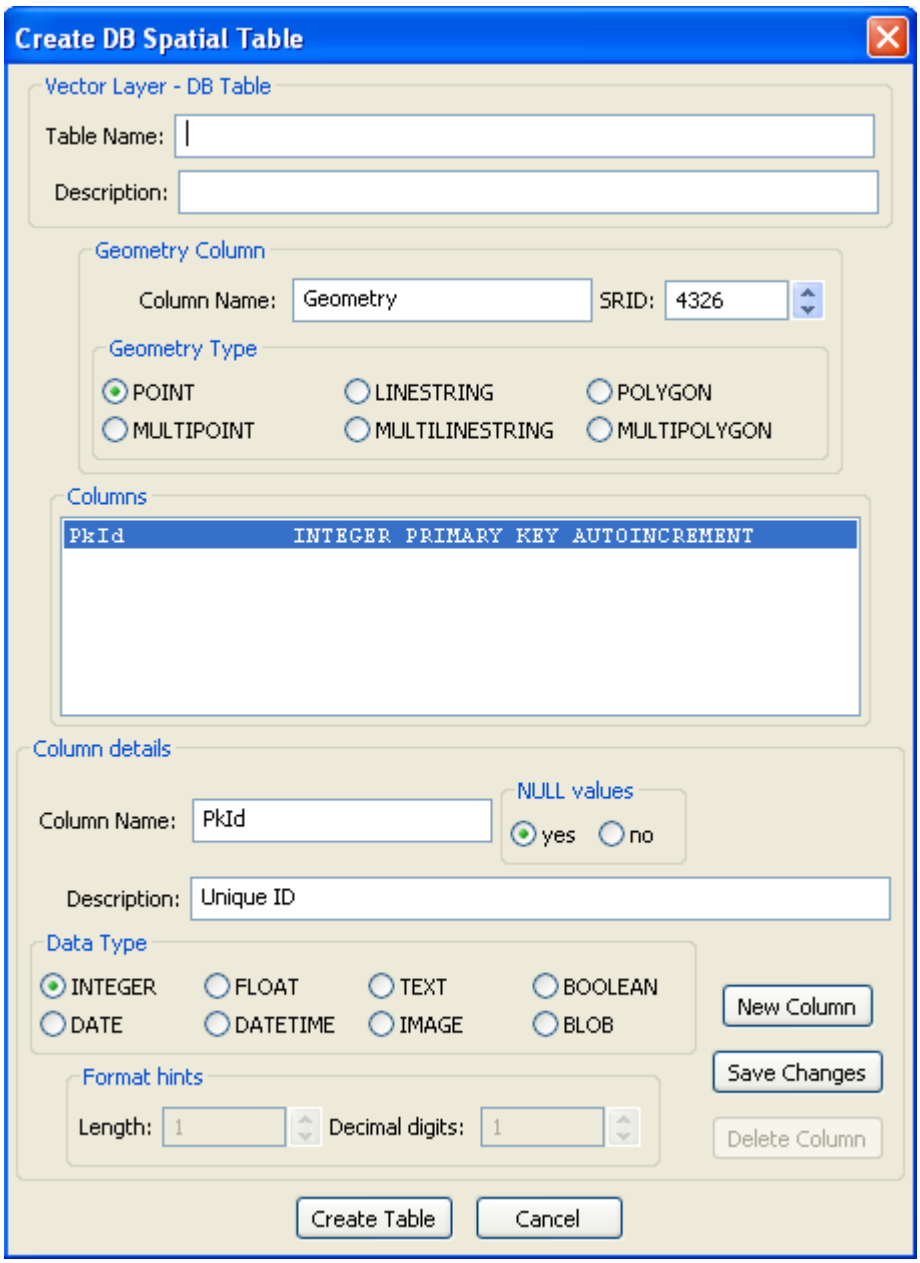


While importing some shapefile you'll be required to select a **SRID** and a **charset encoding** [*may be you already know everything about this, because the companion app* ***spatialite-gui*** *works exactly in same identical way*]

3 – Creating a new DB table / layer



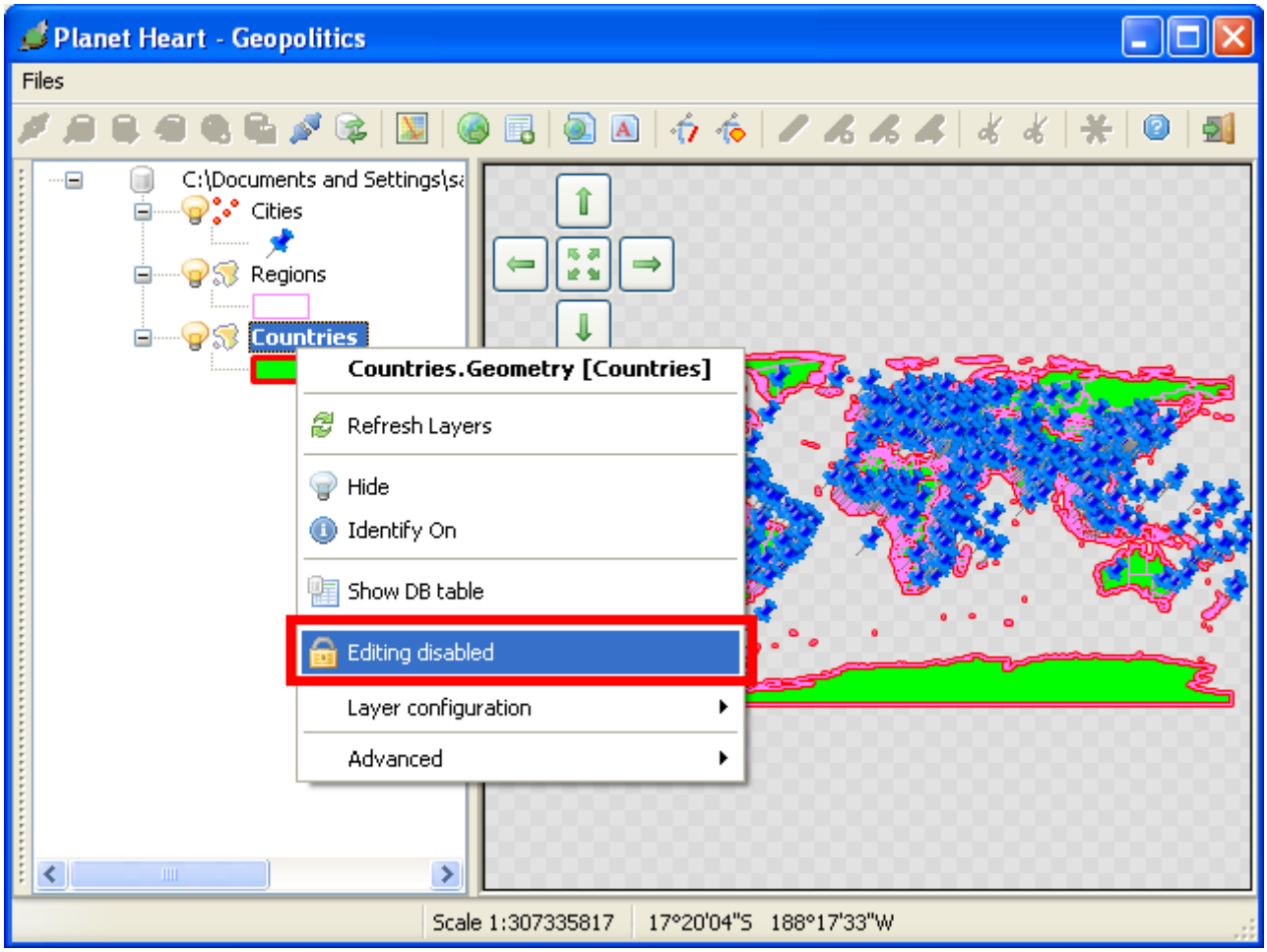
spatialite-gis allows you to create from scratch a completely **new layer** [*DB table*].



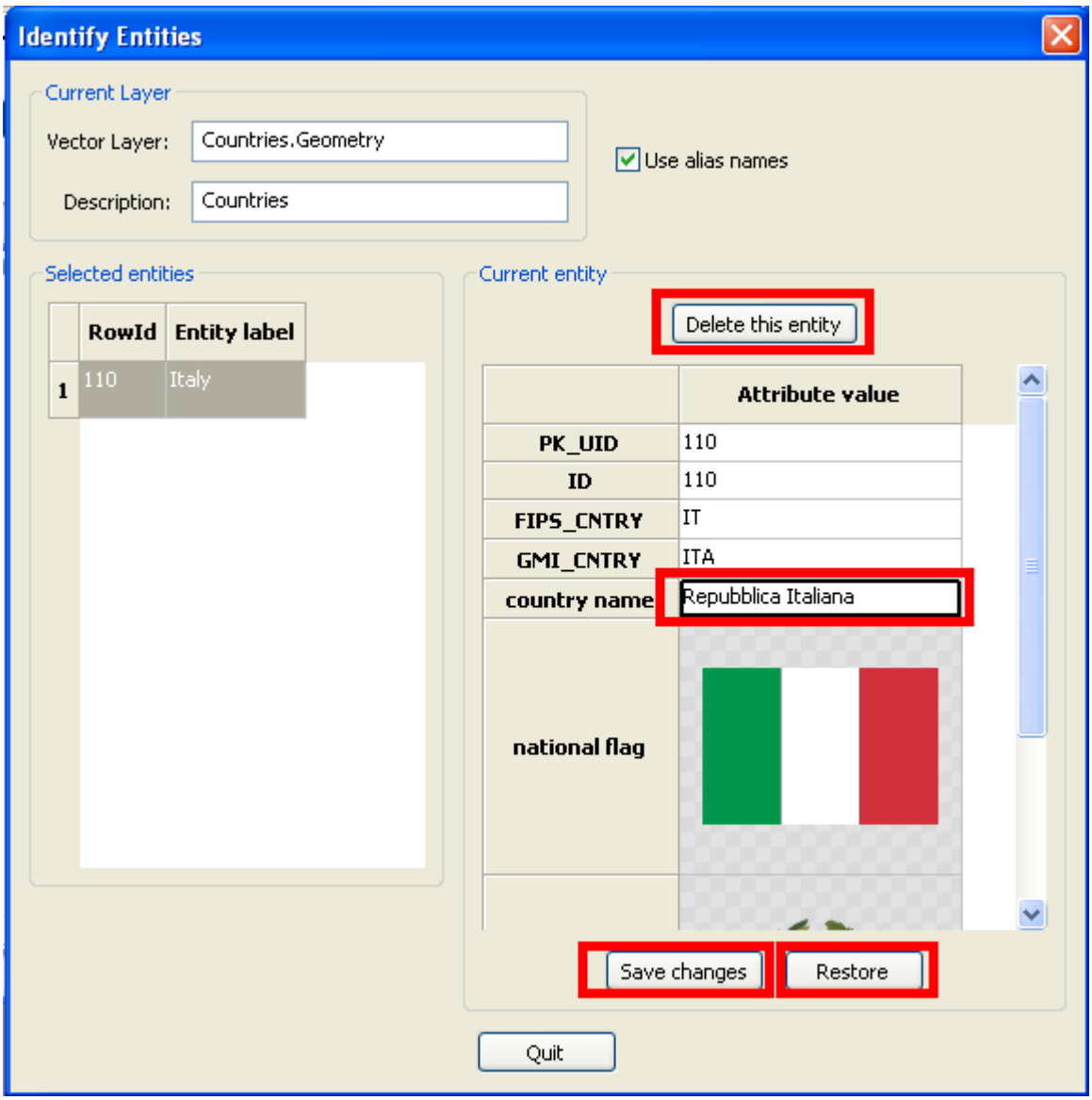
The **create DB table panel** is very similar to the ***table layout panel*** we've already seen while explaining basic operations.
Any Spatial Table found into an SQLite / SpatiaLite DB may be accessed as a layer by **spatialite-gis**
The minimal requirements to define a valid Spatial Table are the followings:

- you've to select a **table name** [*and, may be, an optional human readable name too*]
- you've to select a **geometry column name**, specifying its **geometry type** and **SRID**.
- you've to define a **primary key** of the **numeric auto-increment** type [*this task is automatically performed*]
- then you can freely add any other column [*attribute*] as required. You can defer this task to a later time, because you can use the **table layout panel** to add columns to any already existing table.

4 – Attributes editing

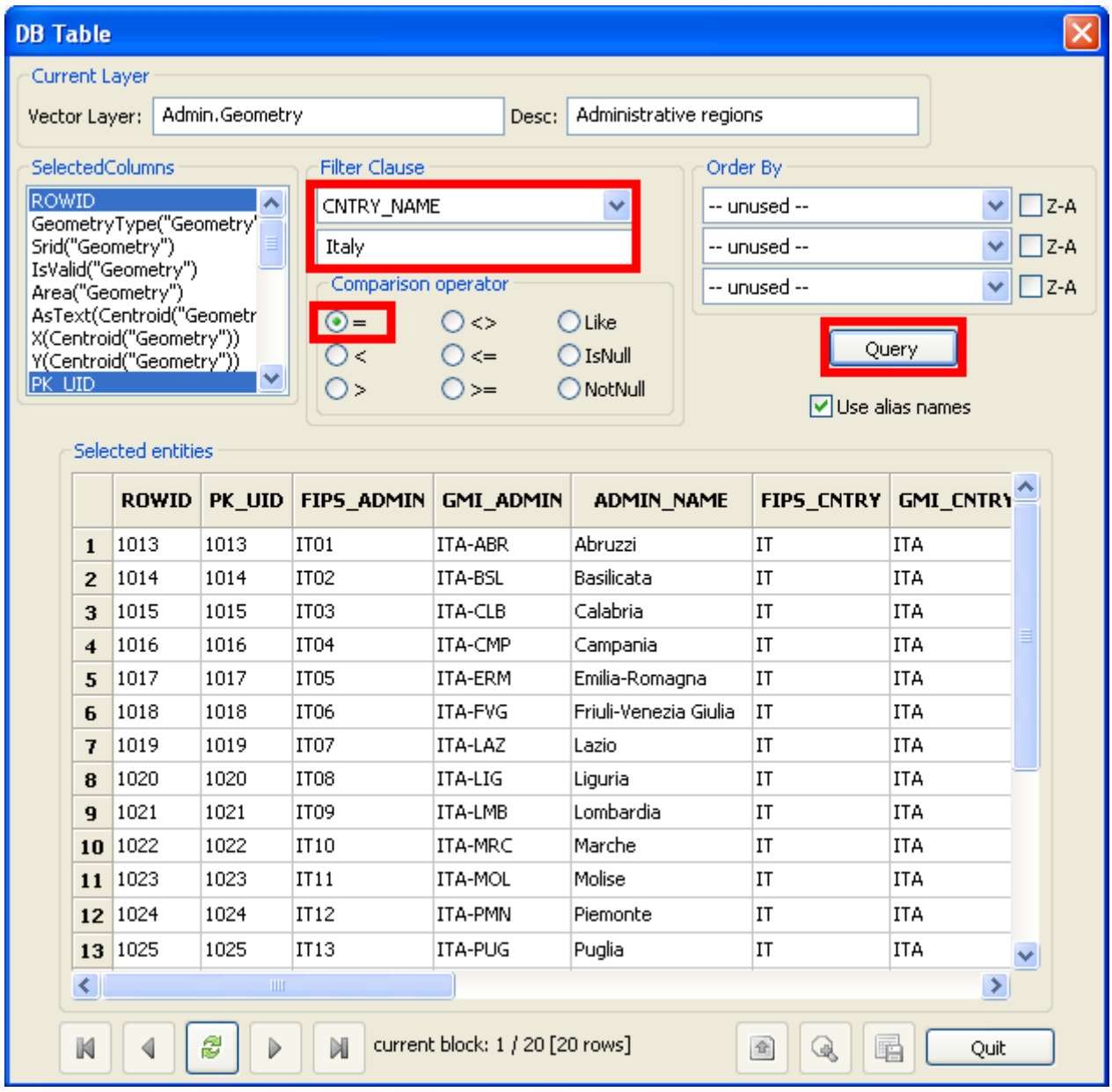


By default **spatialite-gis** disables any editing operation. You have to explicitly enable editing.

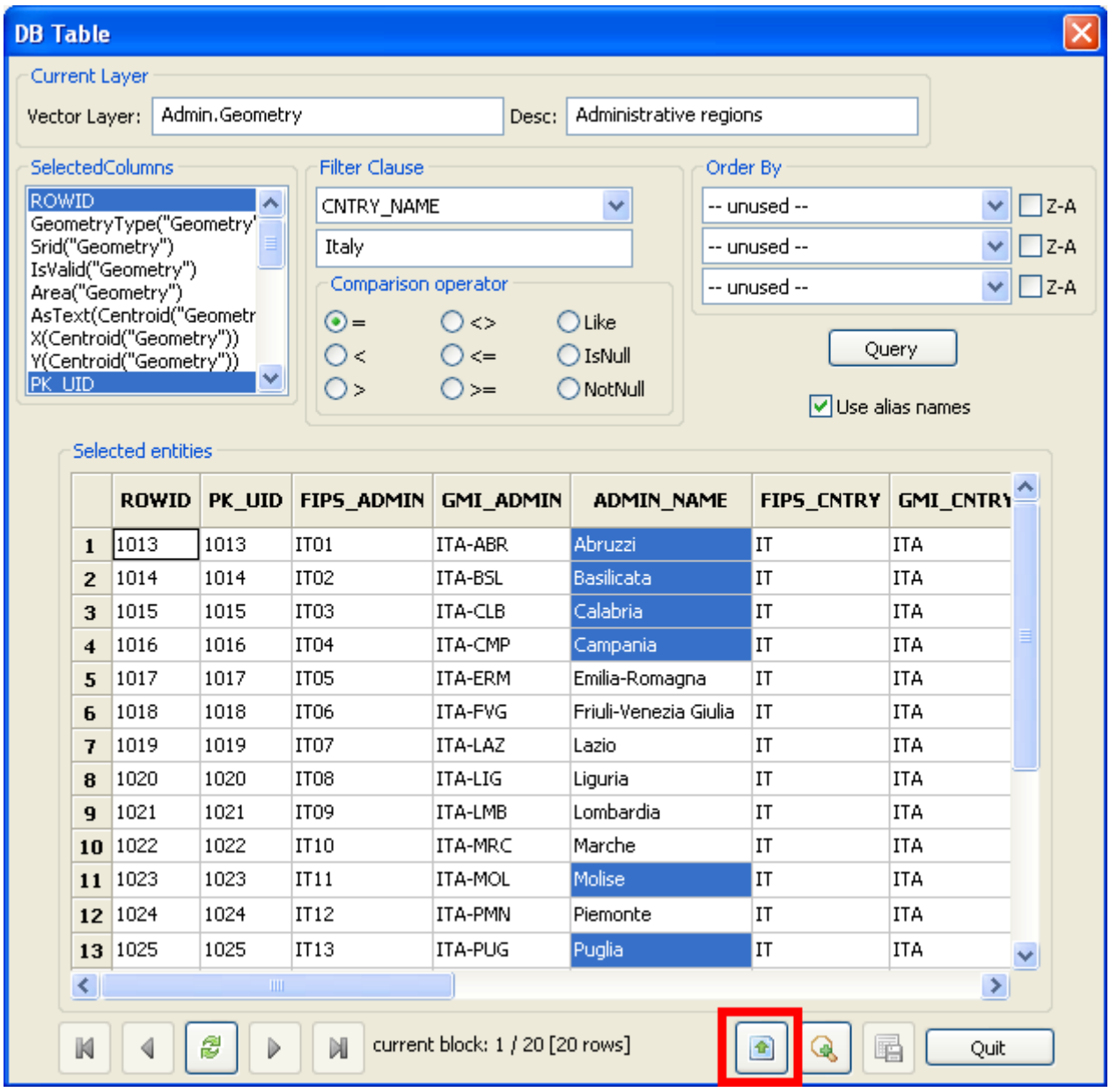


The most immediate way to edit an entity is the one to use the Identify panel.
As you can easily notice, when editing mode is enabled, you are allowed to:

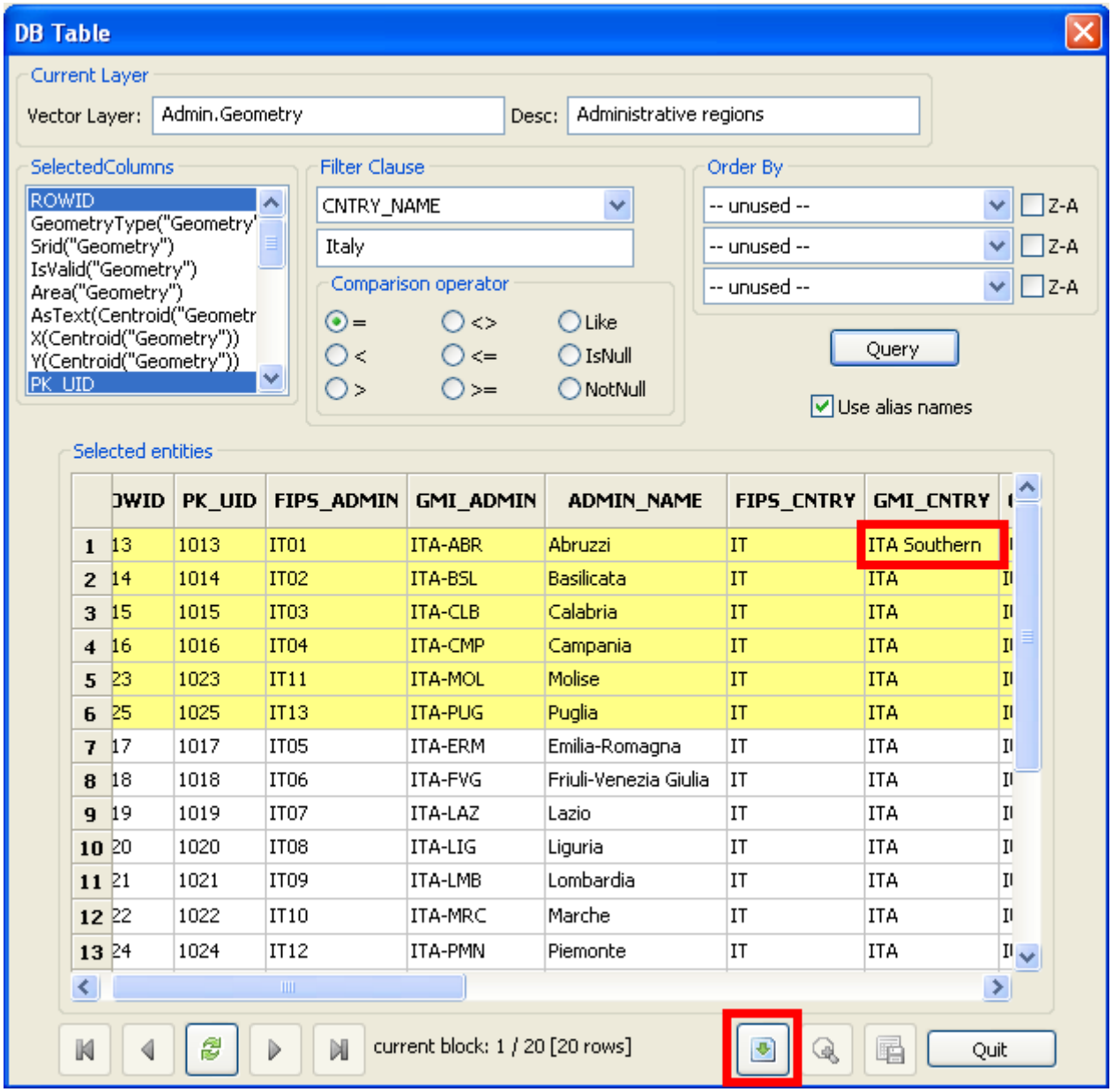
- **delete** the currently selected entity
- **modify** any attribute value
- make any change to a permanent one, i.e. **updating** the corresponding DB row
- and you can recall the original values using the **restore** button [*obviously, before actually saving changes*]



An alternative way to perform attribute editing is the one using the show DB table feature (*obviously, enabling editing mode*).
In this example we have set a filter in order to show only the regions of Italy.

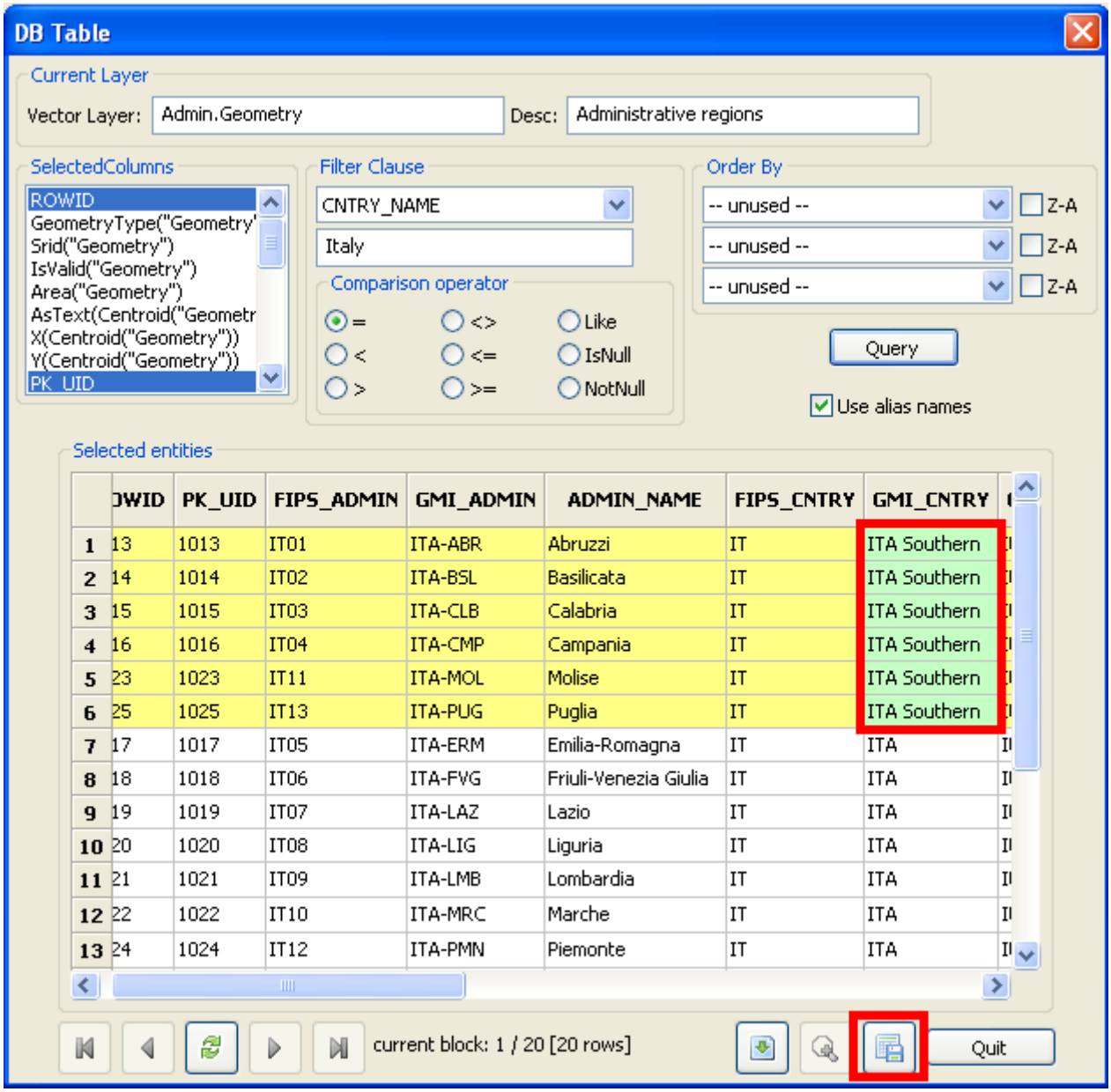


Now we have selected any region belonging to Souther Italy, and then we'll press the **bring to top any selected row** button.



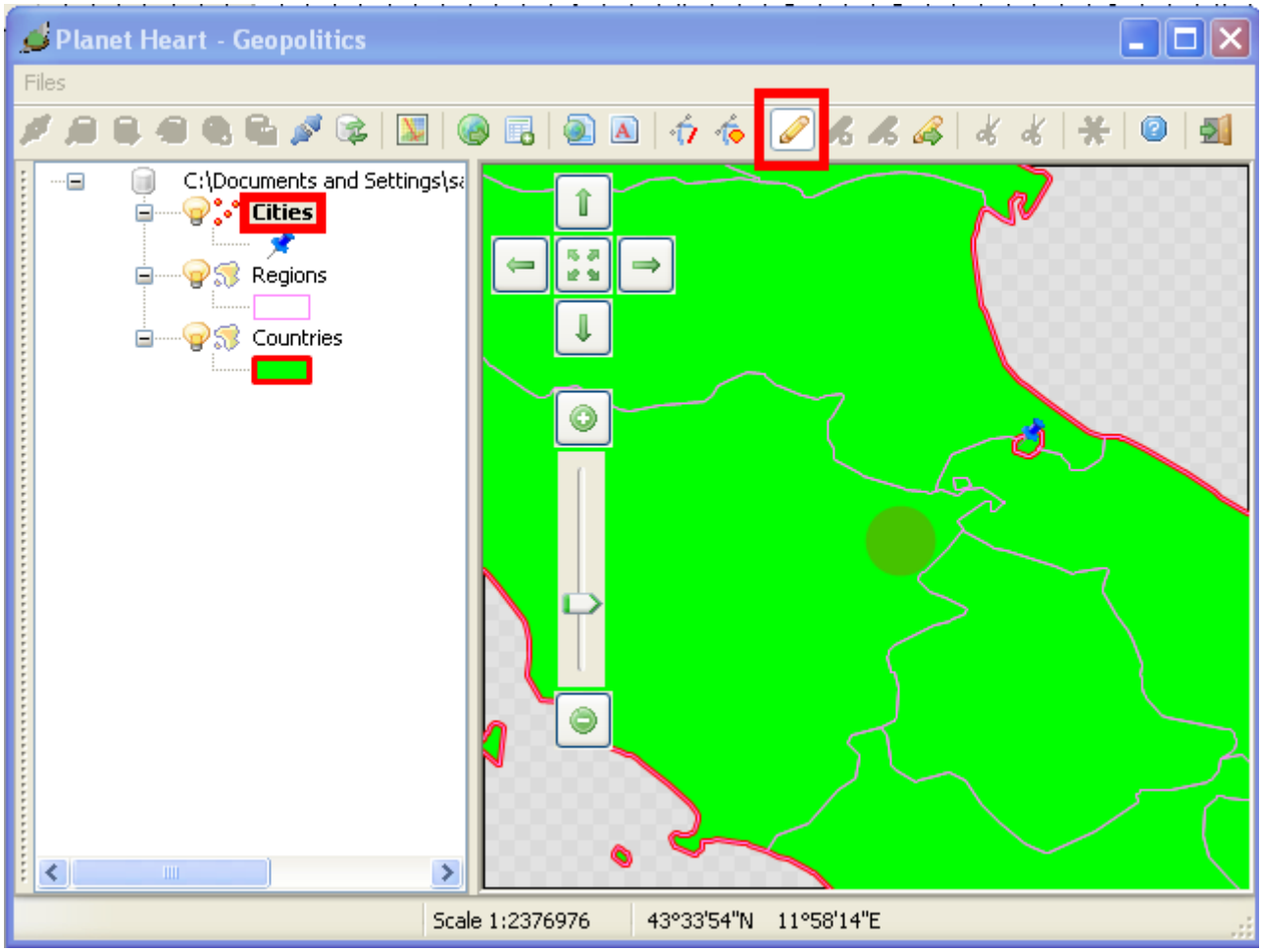
All right: any selected row is now shown on the table top, and the background is highlighted: this actually means any change we'll apply will automatically propagate to the whole selected block.

Please note: if you wish to do so, you can obviously apply a change to a single cell. Simply don't use the selected block at all; you can remove the selected block pressing the **remove rows group** button.



Please note: any modified cell is now shown highlighted: this is because such changes are not yet been saved into the DB, but simply are temporarily cached in memory.
You have to explicitly press the **save** button in order to make any change to be permanently recorded into the underlaying DB.

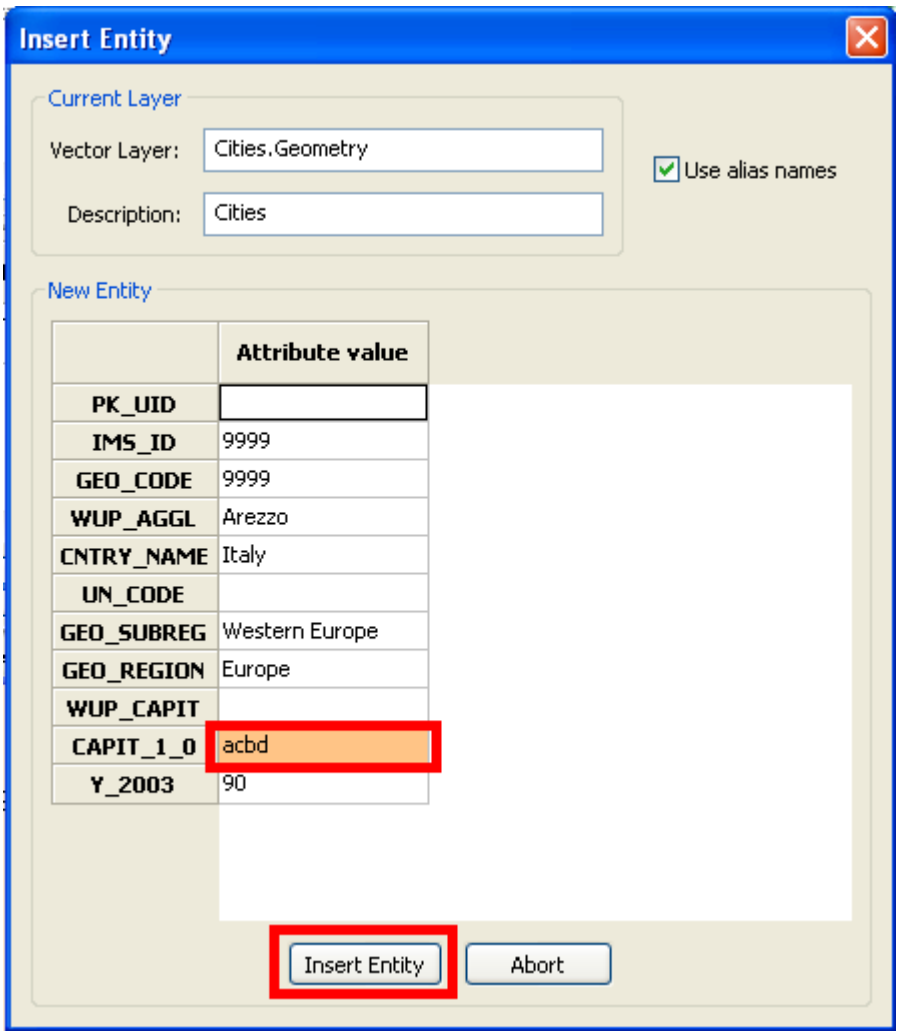
5 - Inserting new entities / drawing



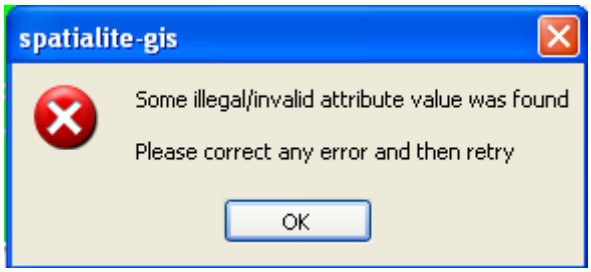
In order to begin a new entity insertion you have prior to:

- make the relevant layer to be the currently selected one
- enable editing operations for this layer
- select the draw tool

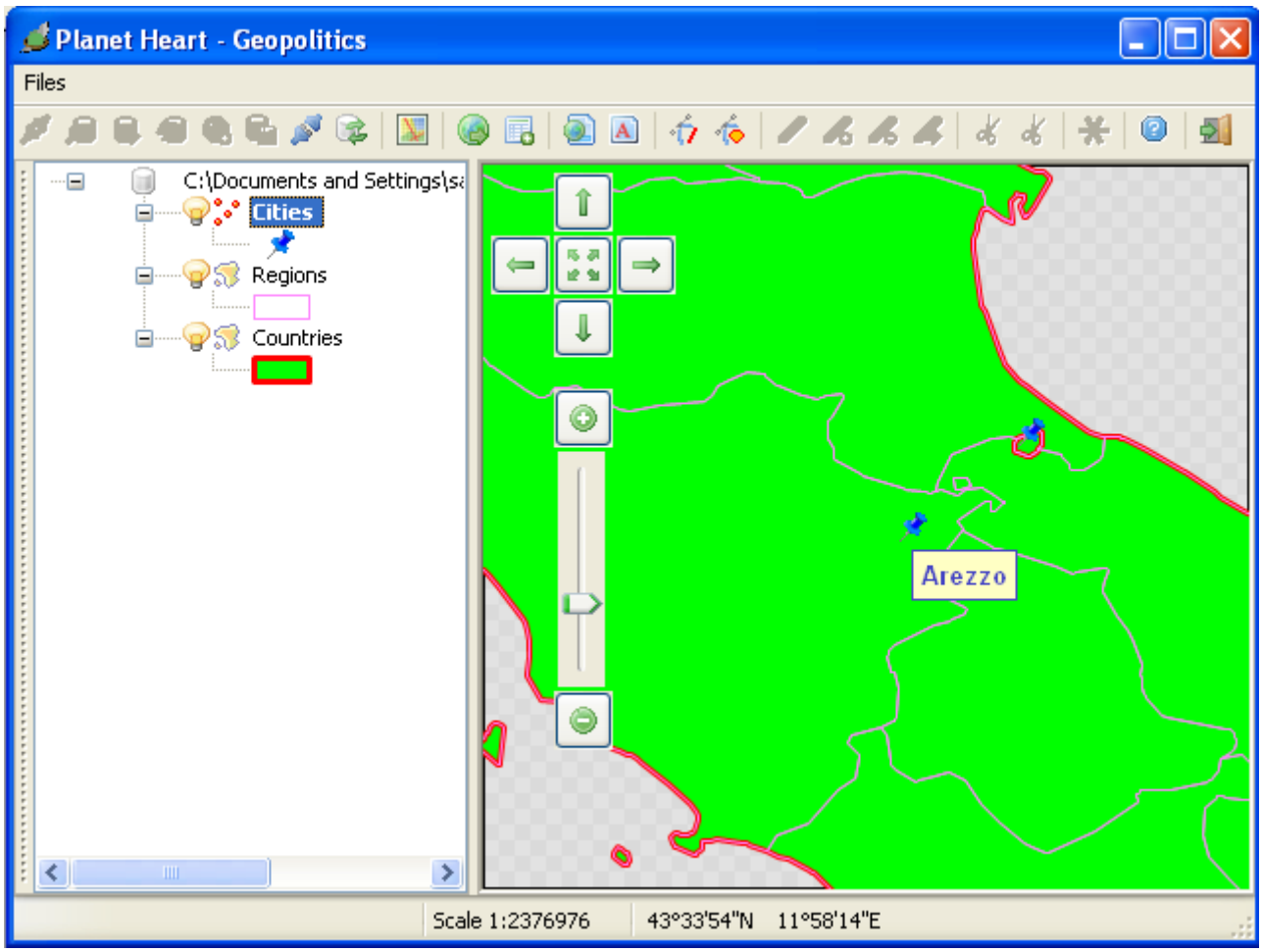
To examine a first example (*the most simple one*), we'll now suppose to insert a POINT-type entity, i.e. a new City.
You simply have to position the mouse cursor over the most appropriate coordinates and then perform a **left mouse button click**: and immediately you'll notice a ***blinking*** point over the map showing the selected point; you've just defined a valid **geometry**, and now you have to assign the corresponding **attributes** in order to complete the new entity insertion.



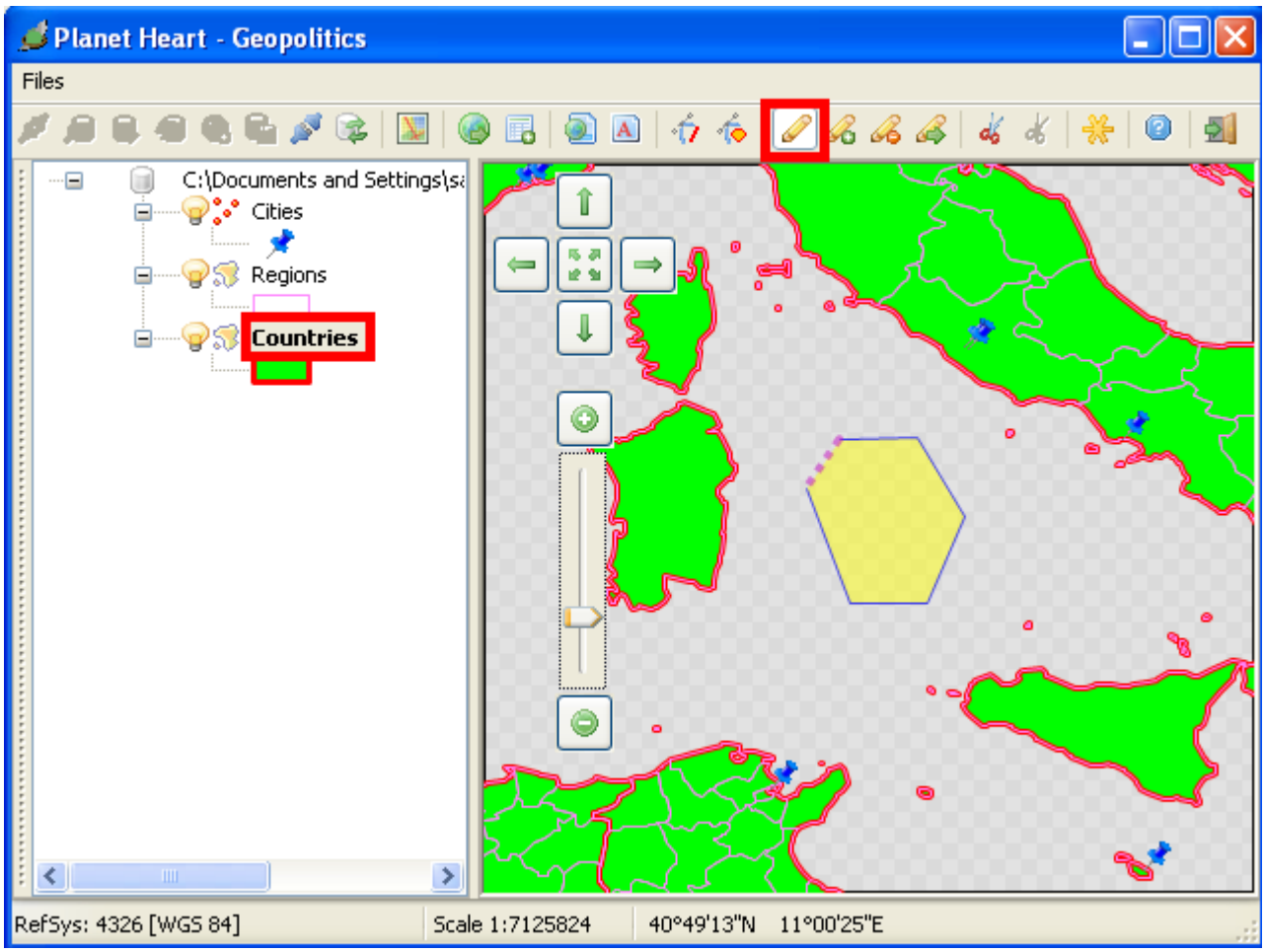
Simultaneously, an **Insert Entity** panel will be shown.
Simply fill any required attribute using appropriate values: if **spatialite-gis** detects any data type incoherency then the corresponding value will be highlighted using a colored background.



Please note: spatialite-gis refuses to insert invalid entities (i.e. rows causing any SQL constraint violation); so you have to correct any issue before actually trying to perform an INSERT.

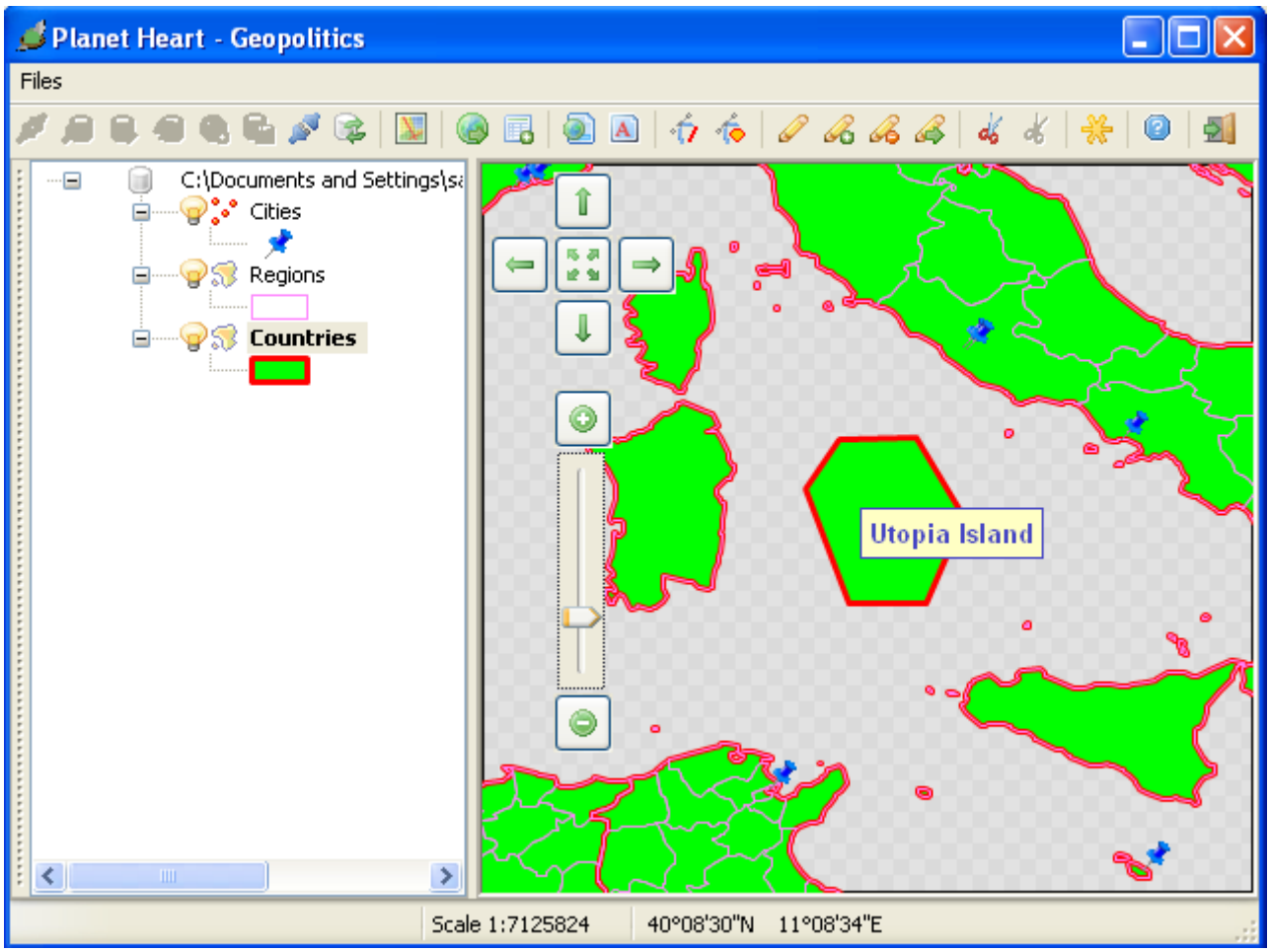


All right: once you've set appropriate values to each required attribute, the new entity you've just now inserted is visible over the map.



As a second example (*a most complex one*), we'll now suppose to insert a POLYGON-type entity, i.e. a new imaginary and fictious Country.
Drawing a polygonal surface is a little bit more complex than defining a single point: procedures related with lines and polygons drawing have already been exposed in the **basic** manual section [*please, re-read the* ***measure polygon tool*** *page if required*].
Simply you have to:

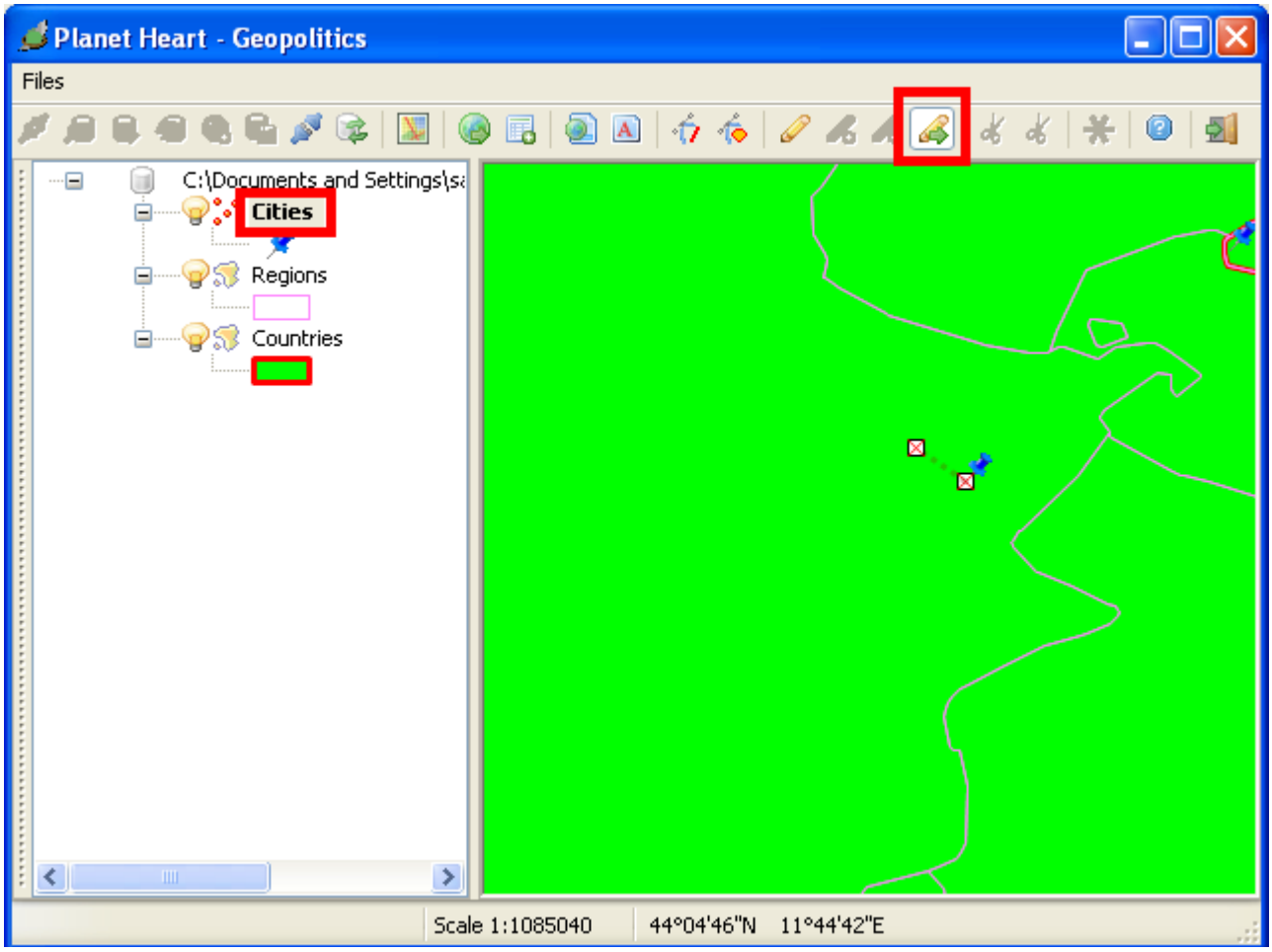
- click the left mouse button for each one vertex you intend to place.
- and then press the END key in order to terminate drawing.
- you can use the CTRL+U key [*UNDO*] in order to discard any wrong or misplaced vertex.



Anyway, inserting a POLYGON entity [and a LINESTRING entity as well] follows the same identical steps required in order to insert a POINT entity:

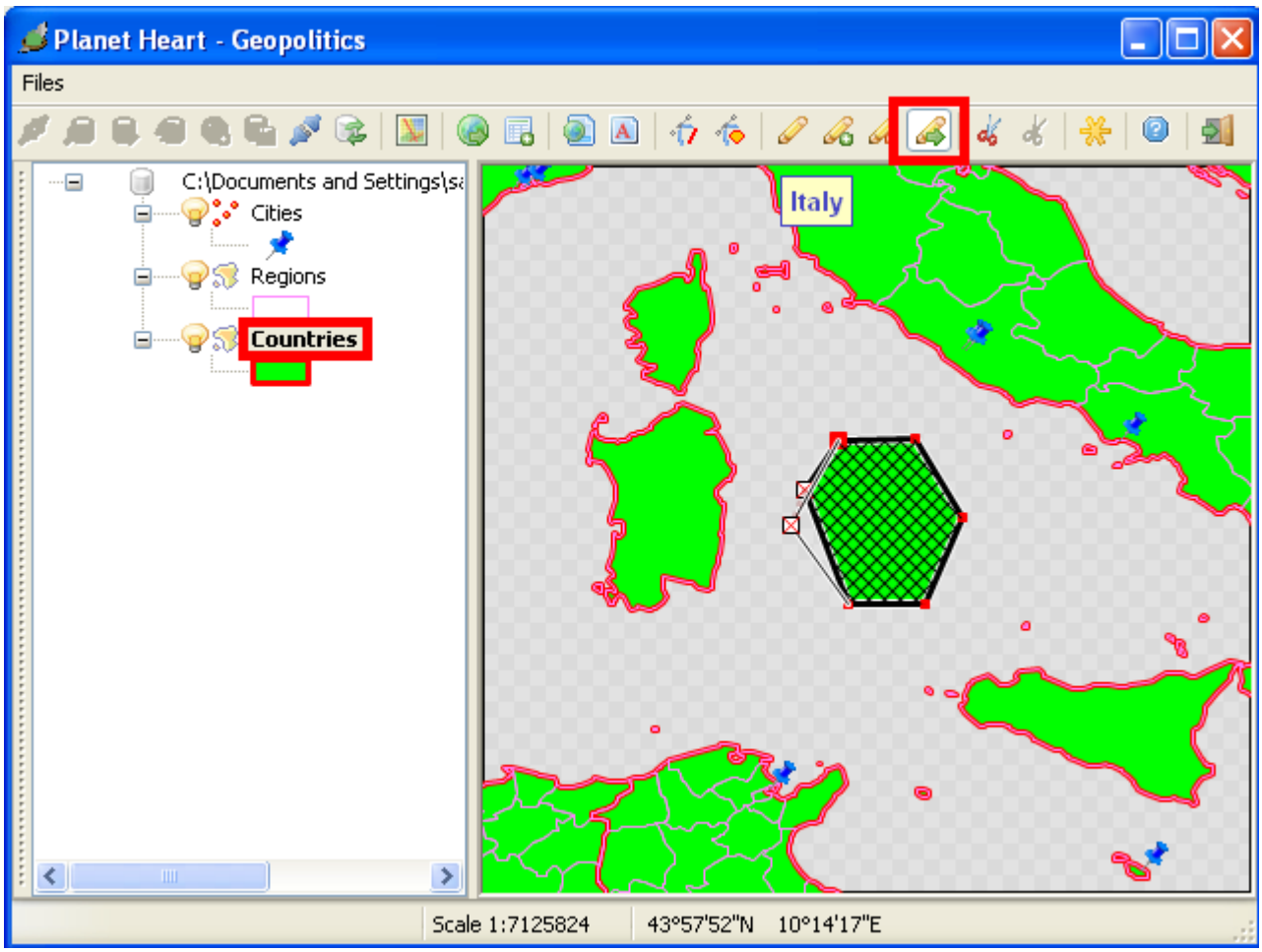
1. first, defining a valid **Geometry** [*drawing*]
2. filling the attribute values using the **Insert Entity** panel
3. once you've completed both steps, the new entity will be immediately shown over the map

6 – Geometry editing: moving a Point or a Vertex



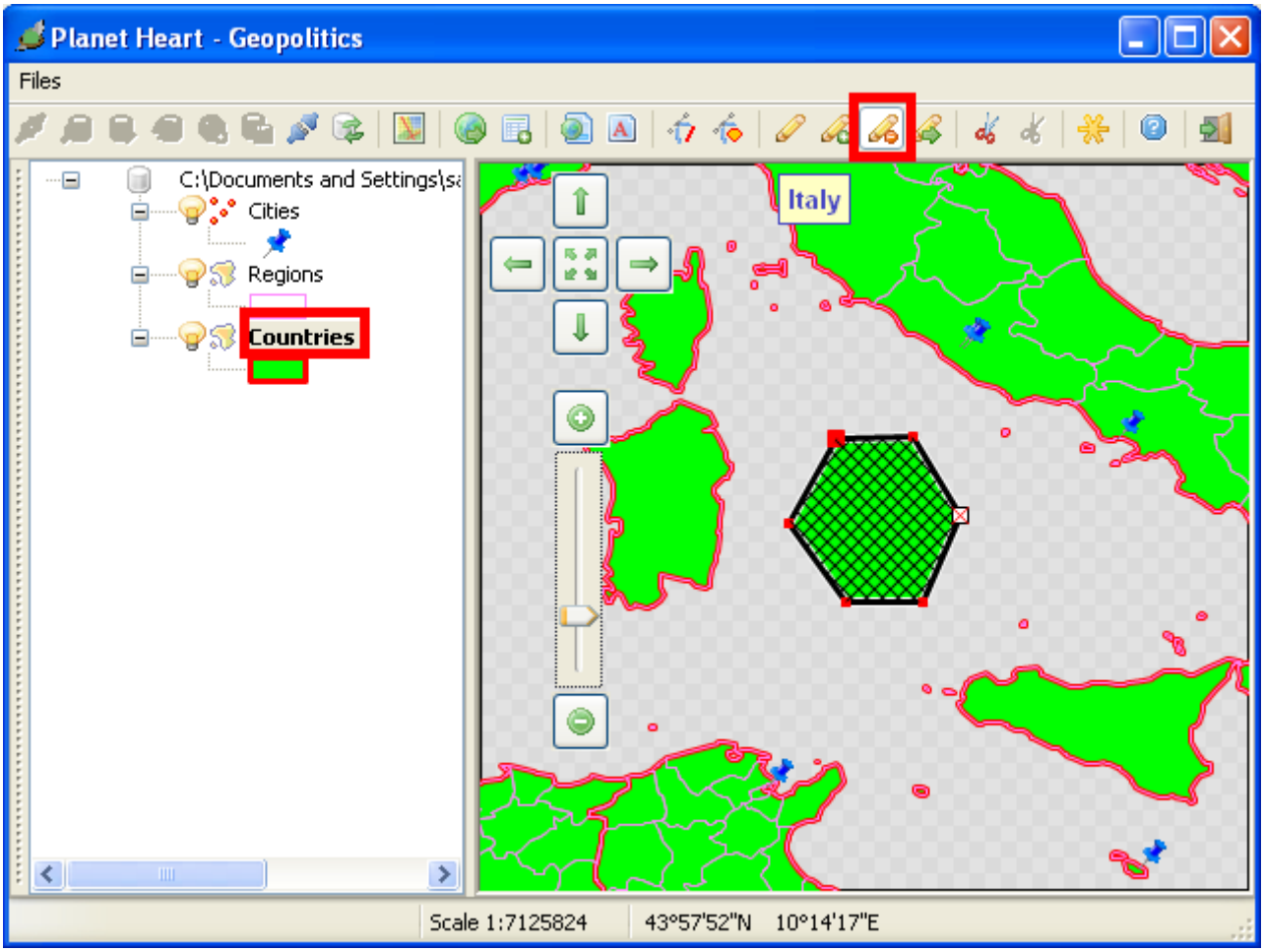
Assigning new coordinates to some already existing Point (i.e. *moving to a different location*) is a very simple task:

1. be sure to set the relevant layer as the currently selected one
2. be sure to allow this layer to be editable
3. select the **move vertex** tool
4. then place the mouse cursor over the Point you intend to move: a special marker will now be shown to confirm this one Point is the currently selected one.
5. now hold the mouse left button pressed down, and start dragging.
6. you'll notice that a second special marker will be shown, following the mouse motion.
7. and in order to make the new position to be permanently assigned to this point (i.e. *completing the move point op*), simply release the mouse left button.



And you can apply the same operation to any Linestring or Ring vertex as well.
In this case you'll immediately notice how the figure changes accordingly to the mouse motion.

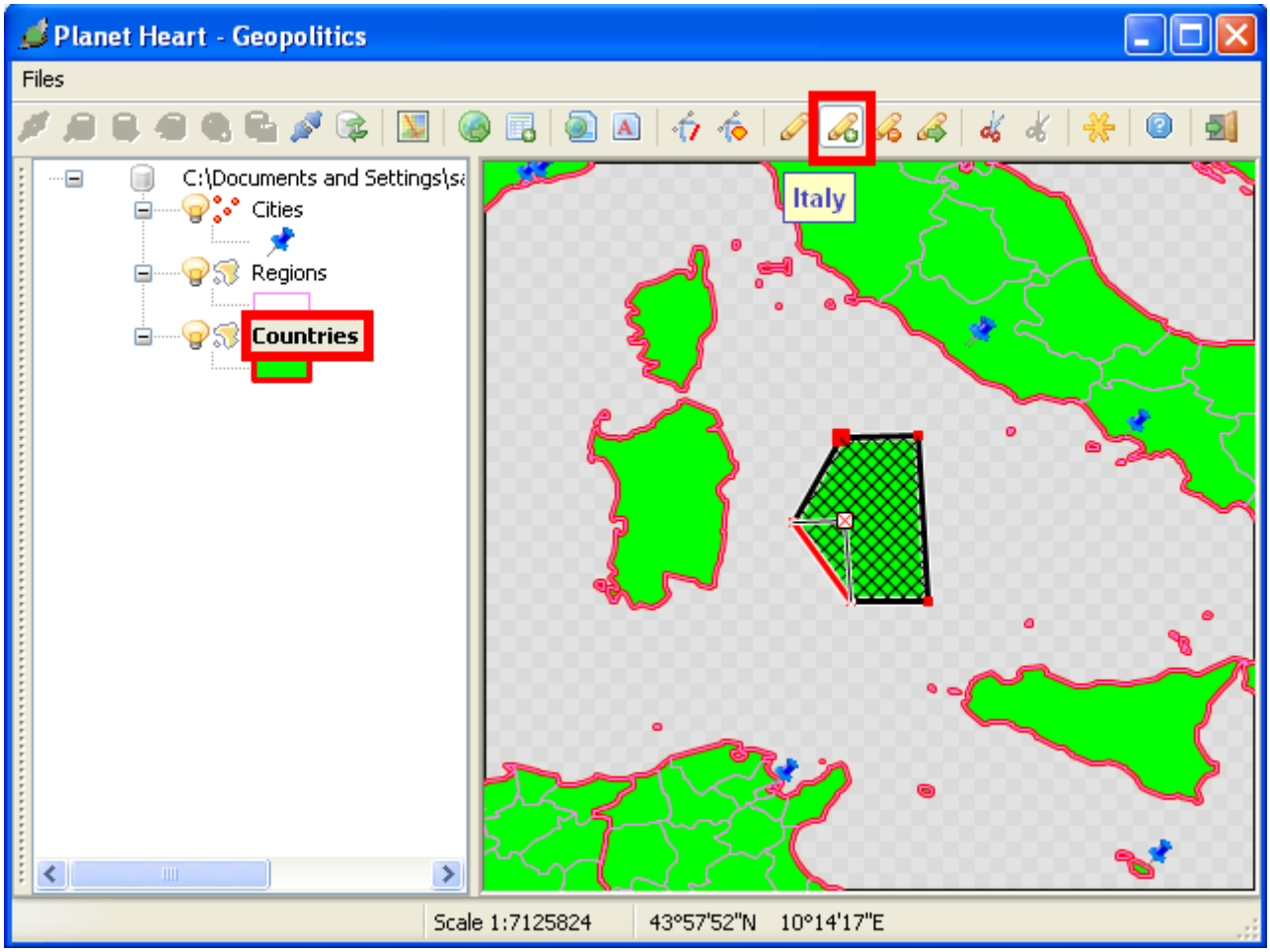
7 – Geometry editing: deleting a Vertex



For Linestring and Ring geometries you can apply another draw tool in order to edit geometries: i.e. the **delete vertex** tool.
You simply have to place the mouse cursor over the specific vertex you intend to remove [*a special marker will be shown to give you a visual feedback*], and then perform a **left mouse button click**, and the selected vertex will be immediately removed.

Please note: under some circumstances, removing a vertex will produce a ***degenerated*** geometry, such as a Linestring containing less than 2 vertices, or a Ring containing less than 4 vertices: in order to preserve anyway geometry consistency and correctness, **spatialite-gis** rejects any attempt to produce such degenerated geometries.

8 – Geometry editing: interpolating a Vertex



And for Linestring and Ring geometries you can apply yet another draw tool in order to edit geometries: i.e. the **interpolate vertex** tool.

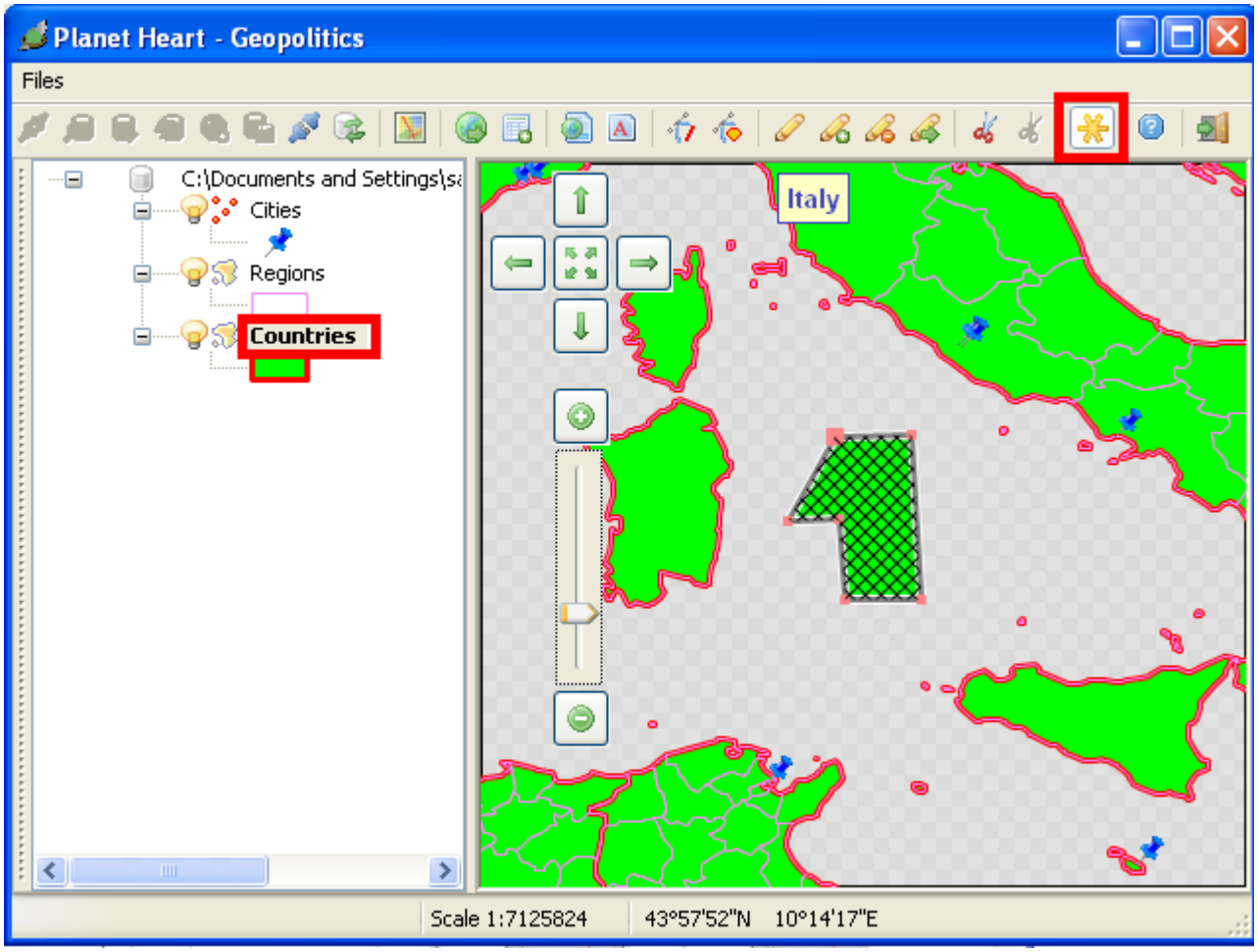
1. place the mouse cursor over the specific Segment you intend to interpolate: a special marker will now be shown to confirm this one Segment is the currently selected one.
2. now hold the mouse left button pressed down, and start dragging.
3. you'll notice that a second special marker will be shown, identifying the interpolated vertex, and following the mouse motion.
4. and in order to make the new interpolated vertex to be permanently inserted (i.e. *completing the interpolate vertex op*), simply release the mouse left button.

9 – Geometry editing: inserting a sub-geometry into a complex Geometry

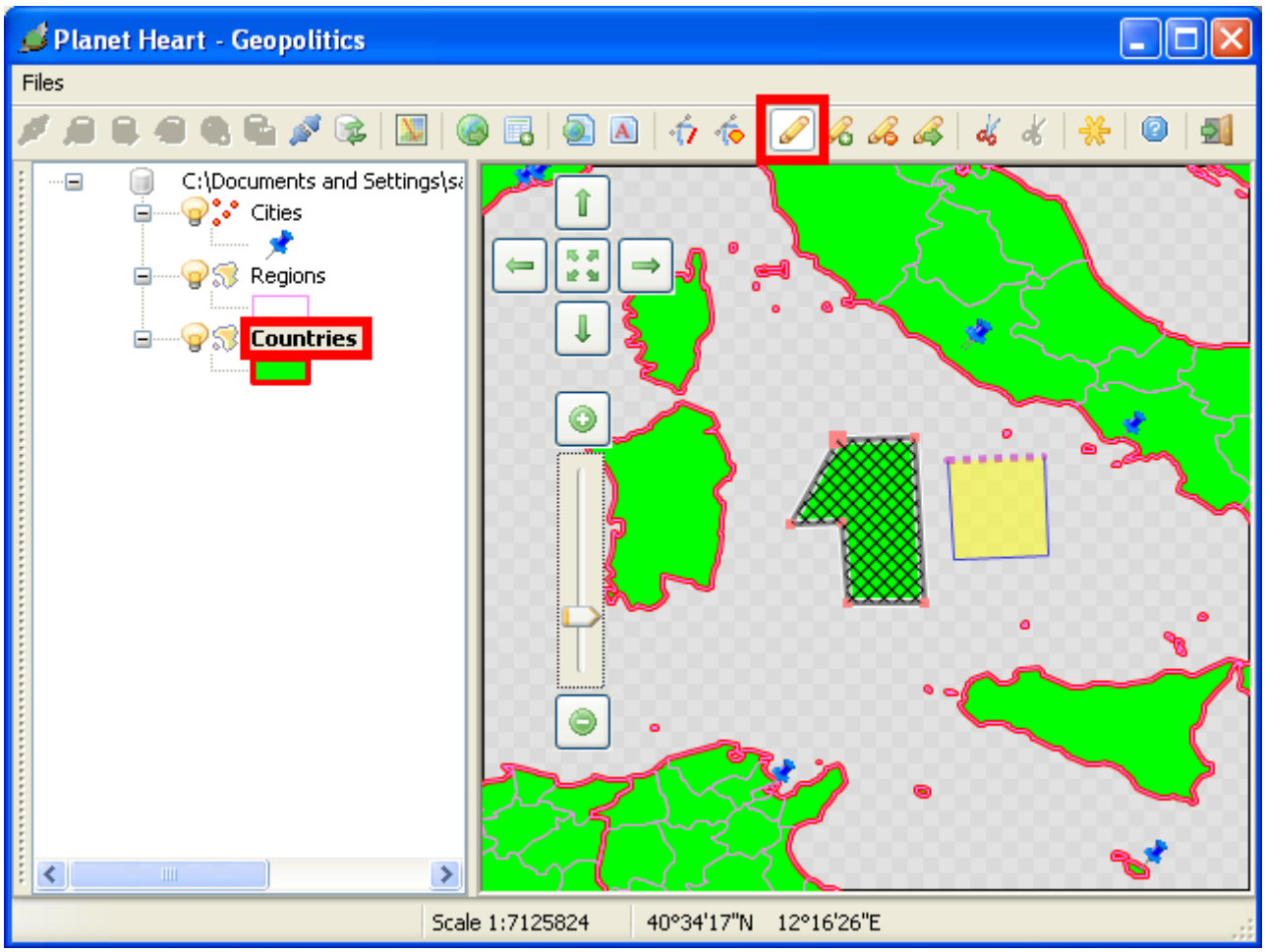
The followings may be assumed to be ***complex*** Geometries:

- a **MultiPoint** i.e. a collection of elementary **Points**
- a **MultiLinestring** i.e. a collection of elementary **Linestrings**
- a **MultiPolygon** i.e. a collection of elementary **Polygons**
- but any **Polygon** as well represents a complex geometry: a polygon is defined by its **Exterior Ring**, but may well be it contains any arbitrary number of **Interior Rings** [***holes***] as well. So, any **Polygon** may be assumed to be a collection of elementary **Rings**.

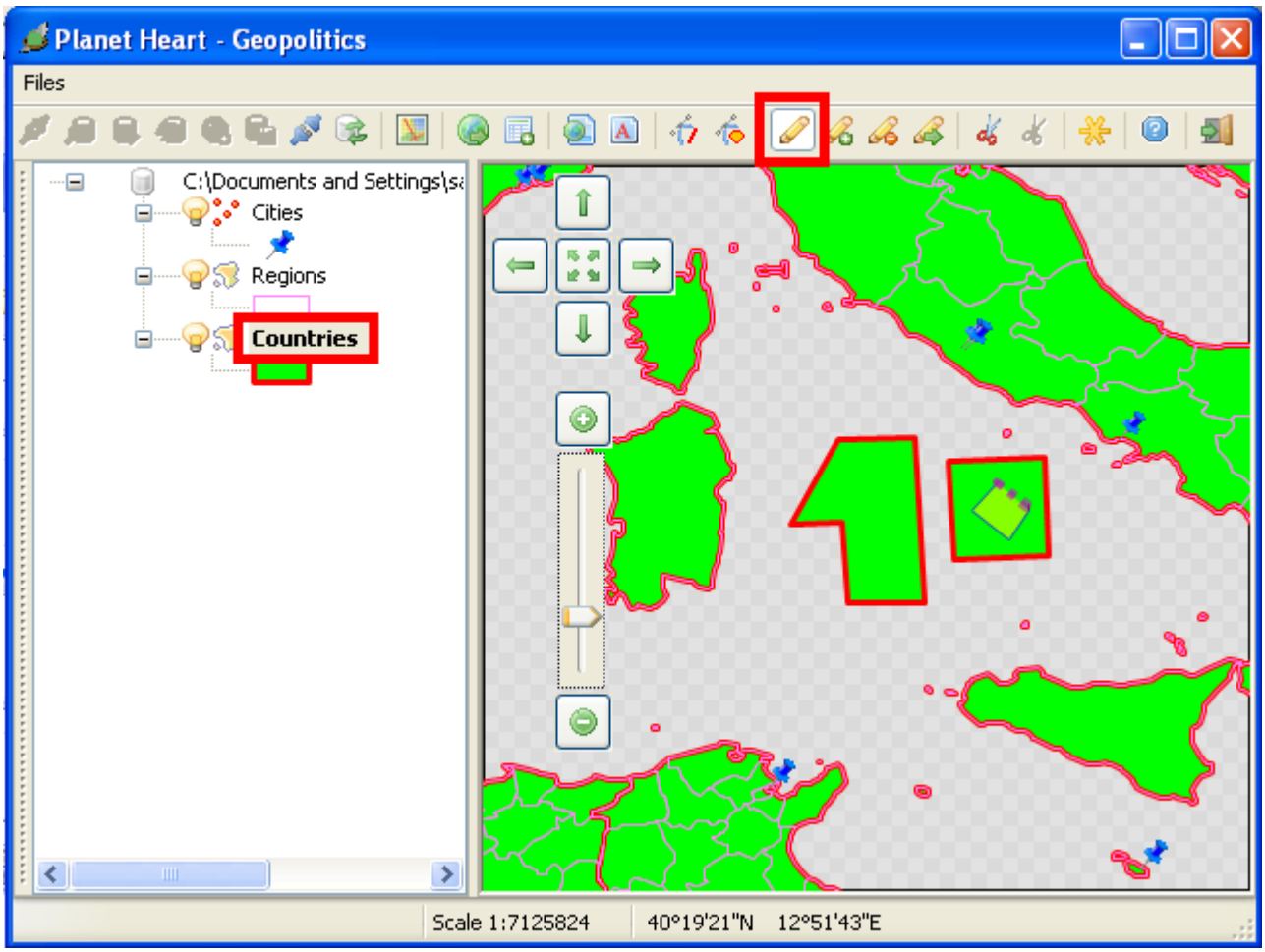
In the following example we'll examine how to add Interior Rings and further Polygons to some MultiPolygon; this one being the most complex case, you'll find really easy adapt by yourself this example to MultiLinestrings and / or MultiPoints.



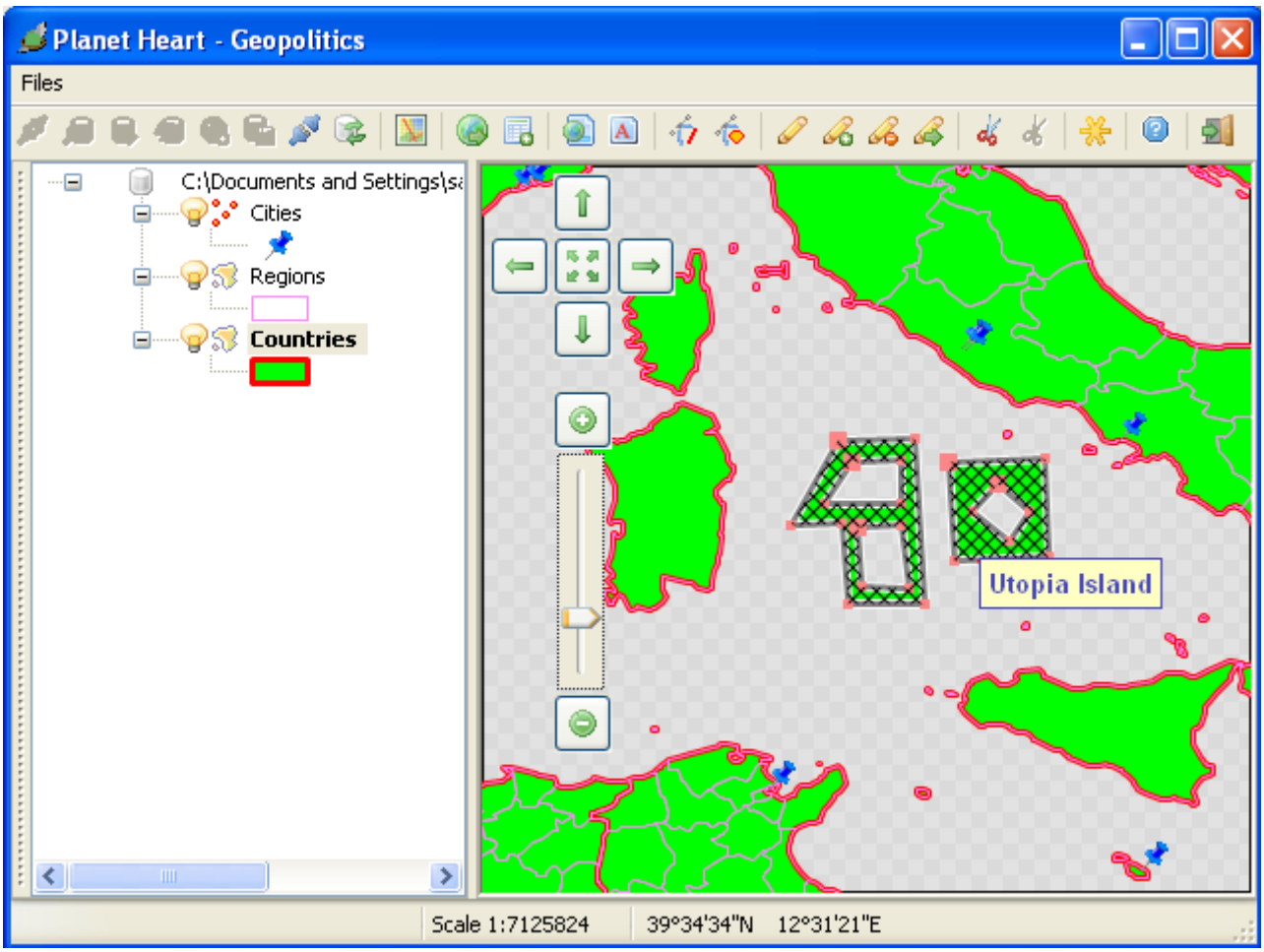
First of all, we are required to **select the current entity**: we can accomplish this task using the **select complex geometry** tool. Actually we'll select the fictious ***Utopia Island*** we've just inserted.
Once an entity becomes the currently selected one, special markers will be shown to highlight this entity over the map.
And when a complex geometry has been selected, then any further drawn geometry will be added to this complex geometry collection, instead of being considered as a new, distinct entity [*as is the standard, ordinary behavior*].



Accordingly to the previously exposed rules, the new Ring we've just now drawn will be immediately appended as a new elementary Polygon, belonging to the Utopia Island MultiPolygon.



And when **spatialite-gis** can automatically detect if the new Ring has to be considered as an Interior Ring, so you can easily add **holes** to any Polygon.



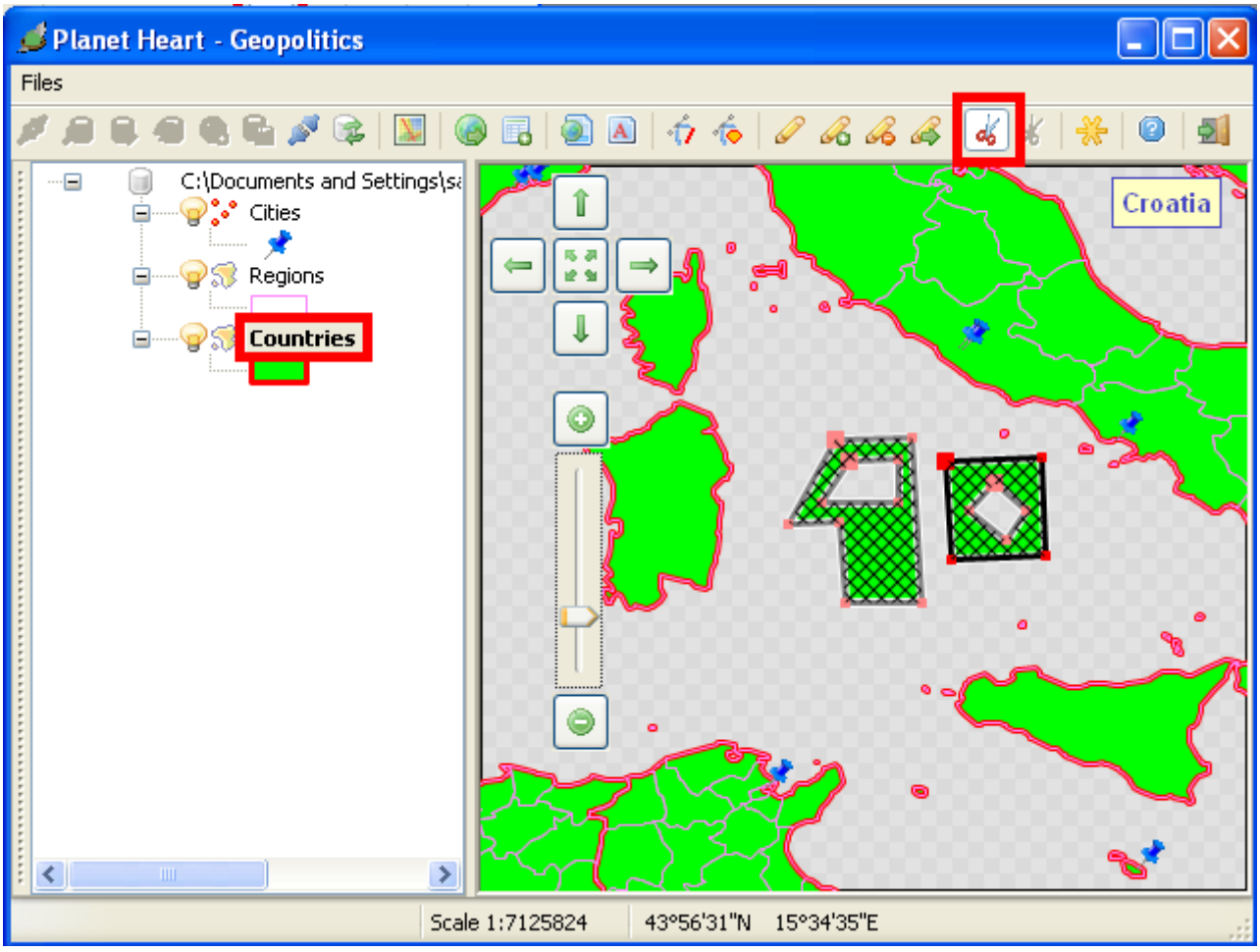
So, in a very few steps, we've been able to define a quite complex geometry, i.e. a MultyPolygon containing two individual Polygons, each one of them containing internal holes.

10 – Geometry editing: removing a sub-geometry from a complex Geometry

In the previous paragraph we've explored how to use **spatialite-gis** in order to add new elementary geometries into a complex Geometry.
Quite obviously, the opposite operation is supported as well, i.e. the one to remove elementary geometries from a complex Geometry. This includes:

- removing a **Point** from a **MultiPoint**
- removing a **Linestring** from a **MultiLinestring**
- removing a **Polygon** from a **MultiPolygon**
- removing an **Interior Ring** from a **Polygon**

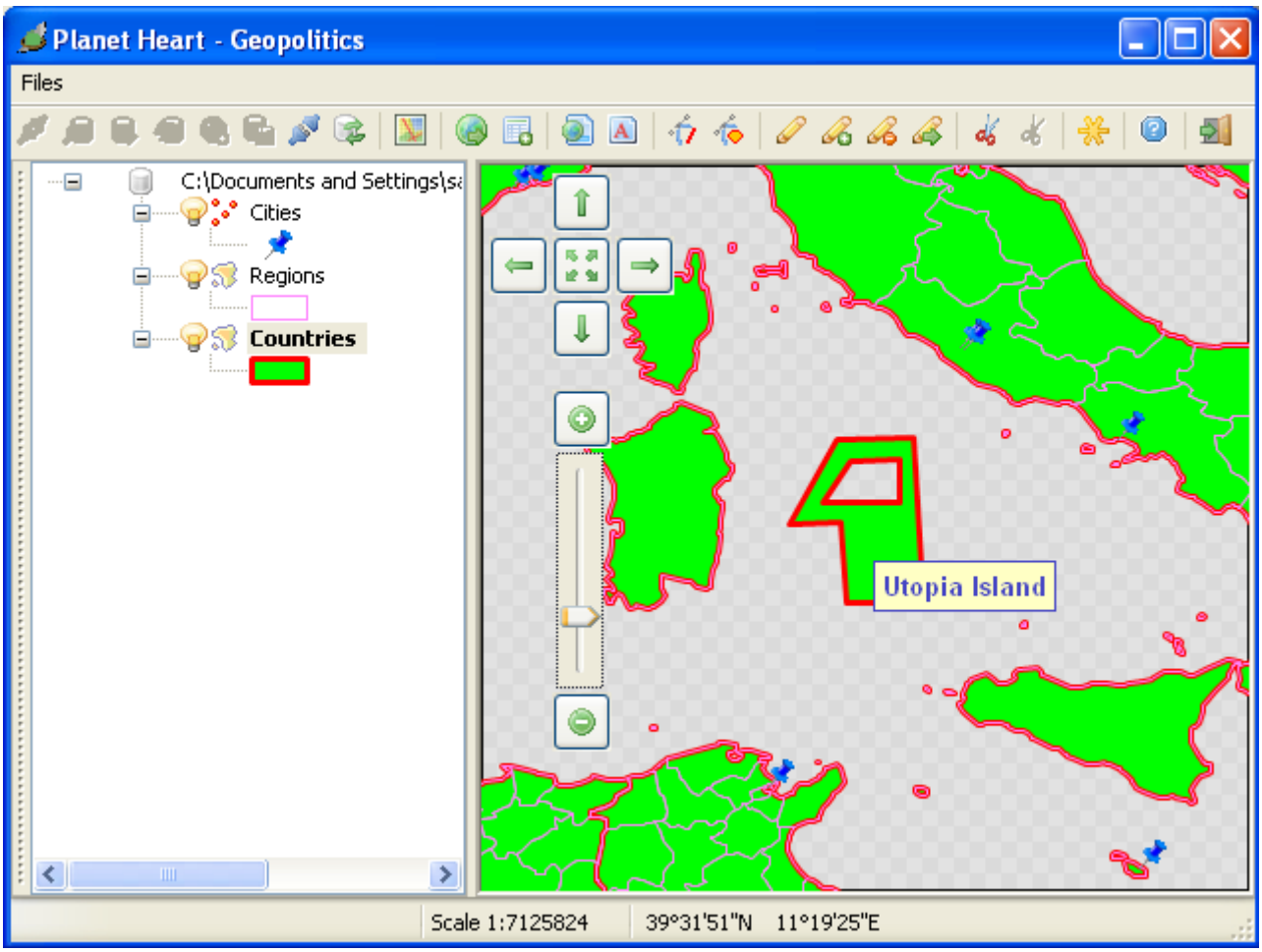
Please note: under some circumstances, removing an elementary geometry from a collection will produce a ***degenerated*** geometry, i.e. a **NULL** [***empty***] one: in order to preserve anyway geometry consistency and correctness, **spatialite-gis** rejects any attempt to produce such degenerated geometries.



We'll continue using the fictious ***Utopia Island*** in this example.
You simply have to select the **delete sub-geometry tool**, and then select the individual ring you intend to remove; special markers will be shown to highlight this entity (and the specific item) over the map.
In order to confirm deletion of this item, you simply have to perform a **mouse left button click**.

In the special case of polygon's rings, the following rules apply:

- Interior rings [holes] will be removed on an individual base
- Deleting an Exterior ring implicitly assumes deletion of the whole Polygon [*this including any interior ring as well*].



And in this case too we've been able to edit a complex geometry in a very few and simple steps.