DEVELOPING AND IMPLEMENTING MULTIUSER, FULLY RELATIONAL GIS DATABASE FOR DESKTOP SYSTEMS USING OPEN SOURCE TECHNOLOGIES

Zsolt MAGYARI-SÁSKA[1]

ABSTRACT:
At present GIS tends toward two directions: desktop GIS and on-line GIS. On-line GIS is mostly for data browsing while desktop GIS for research. But working on shared datasets is a common concept for nowadays data processing, and is useful also for researchers. That's why in our publication we try to present an open source way to implement a desktop based GIS, but which can work over shared data and which offers more power on database management than simple desktop GIS can.

Key-words: *relational database, data integrity, shared dataset, handy tool, custom user interface, open source software, QGIS, PostGIS*

1. INTRODUCTION

One of GIS subsystem is the database module, which should provide data storage and database management functionality. Even that this is a declared scope, the current GIS doesn't fulfill this requirement. The relation between spatial elements and the corresponding data tables are working well using efficient methods maintaining the correspondence with each other, but all descriptive data are stored mainly in one table per layer.

However there are possibilities to include other, not layer related tables in geodatabase. With these tables limited4 relations can be defined, but these options provide only a limited one-to-many relationship model, key values are not verifiable and the many-to-many relationship is not available.

These deficiencies of current GIS are serious drawbacks both for scientist and common users. A scientist should not know the details of a relational database and should not spend energy on maintaining the database consistency. This is true also for any common user of a GIS project. The GIS project developer should offer these functionalities, which are not fully implemented in current GIS.

Effective data storage and data management is one of the main issues of GIS. The other one, not taking account the analyzing functionality, is the user interface (UI). The UI for data manipulation should facilitate the workflow, by providing common, well known graphical elements to enter, select, modify and select interrelated data.

Current GIS provide some, but limited functionality in this direction. Many of them makes possible to generate a form based on table columns, but without modifying the fields label or changing the entry object type.

A most common way to create a user interface for a multiuser GIS is to transfer it to an online platform. This is a viable method, but by this way we lose the main force of GIS, the possibility to perform complex analysis.

[1] *Babeş-Bolyai University, Faculty of Geography, Gheorgheni, Romania, zsmagyari@gmail.com*

In this article we propose to present a possible implementation of a multiuser GIS project, in which the data storage and management respects the relational database implementation standards, is worldwide available through Internet, has a user defined interface and is accessible in a desktop GIS, so all analysis functions are available and also to have high quality data transfer and processing (Nguyen, 2009).

2. INITIAL SPECIFICATIONS

2.1. Technical specifications

Open source solutions in GIS tend to be more and more effective and widely used (Gerlek, 2007). Any freely available robust, comprehensive system would spread rapidly between users, that’s why we proposed QGIS as desktop GIS for our project (Lemmens, 2008). This fact is proved by several very well realized project with open source software (Gkatzoflias et al., 2003)

For database management we used PostgreSQL with PostGIS extension to store and handle the GIS database, having all the necessary functionalities to maintain database consistency (Marquez, 2005, Corti et al., 2014). Apart from these PostgreSQL data access and user management part, in this way the security issues for a multiuser environment were solved (Chitj, 2015).

The user interface for descriptive data management was developed using Qt4 Designer.

All mentioned elements: database handling with user management, GIS analysis and user interface development are based on open source and freely available but efficient technologies, granting an easy access without losses in functionality.

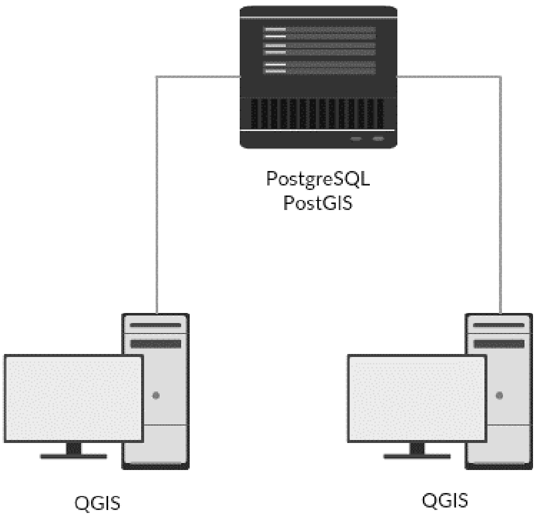


Fig.1 - Client-server system architecture, with used software

Even if there could be several system architecture, we will have one computer as data server and multiple GIS clients (fig. 1). The number of concurrent users will be limited at 20. We also want to define 3 user levels: administrator, data manager and data user. Administrators will have all rights including user management, data managers will have the right of modifying data, while the simple users can only visualize the data.

2.2. Model specifications

To illustrate all necessary phases to obtain a real life related project, we choose the following example.

We want to realize an inventory regarding amphibians. We want to mark the geographical positions where they were seen, recording the species name and the threat on them along with the severity of the threat.

After the specific database design steps the following scheme describes the database (fig.2).

We can observe one-to-many and also many-to-many relationship between the data tables. A species can be recorded at multiple positions, a threat can affect several amphibians, while an amphibian can be affected by several threats.

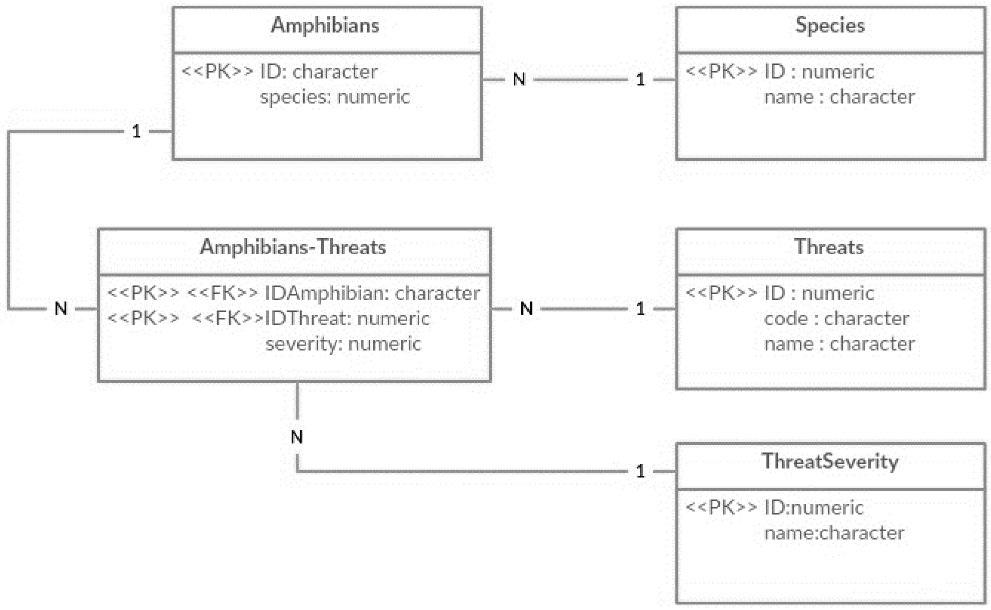


Fig.2 - Example database UML diagram

3. IMPLEMENTATION

Being a multiuser project it's obvious that should have server and client side components. In our project all data are stored at server side, but handled on client side in a desktop GIS. Server side components are PostgreSQL with PostGIS, while client side component is QGIS.

3.1. Server side

Depending on the used operating system the PostgreSQL installation may vary, but the download kit is available through its homepage (www.postgresq.org). The PostGIS extension which extends PostgreSQL to can handle spatial databases is available at postgis.net but also through StackBulider application in case of Windows OS.

After a proper installation access methods should be configured. For this `pg_hba.conf` and `postgresql.conf` can be modified (Stone, Matthew, 2007).

To satisfy our proposed limitation the following changes has to be made

- maximum 20 concurrent users: in `postgresql.conf` the `max_connections` parameter has to be set to 20
- data access from external client with IP address 192.168.1.100: in `pg_hba.conf` a new a line has to be added with `hostssl model all 192.168.1.100/32 md5`

If we would develop a web application on the same server where the PostgreSQL and database are, a localhost access would be sufficient.

The pgAdmin III (local) or phpPgAdmin (web) utility are useful to define the main characteristics of the database, including table structures, relations, users and their rights over datatables and possible operations (Schönig, 2014).

In pgAdmin III to view all users the Show user for privileges options should be selected from Options. When adding a new user (role) we can specify if the user can create new users or if he can create new database. In our case the just the administrator will have such rights. To assign other, default privileges, regarding different operations over tables (select, update, delete, insert) is also possible at this point, but it can be modified for each data table later.

Adding the PostGIS extension to the database is possible in Extension section.

When creating a new table you have to pay attention to the following issues:

- every table should have a primary key, which can be a serial type column or a user filled column.
- the geometry type column indicates, that the table is in fact associated to a layer
- using pgAdmin III, you cannot specify geometry type and coordinate reference system, this should be done manually
- you cannot add SQL code to create data table while you don't specify the table's name
- geometry fields with coordinate reference system cannot be added interactively

For example to create the Amphibians table, you could went through the following steps:

- create a new table in pgAdmin and type any name
- insert the proper code in SQL tab (deactivate `ReadOnly` checkbox)
- later new fields can be specified interactively using pgAdmin

```
CREATE TABLE "Amphibians"
(
 ID character varying(15) NOT NULL,
 geom geometry(MultiPoint,31700),
 species integer,
 CONSTRAINT "Amphibians_pkey" PRIMARY KEY (ID),
 CONSTRAINT "Amphibians_fkey_1" FOREIGN KEY (species)
      REFERENCES "Species" (ID) MATCH FULL
      ON UPDATE CASCADE ON DELETE NO ACTION
);
```

Fig. 3 - SQL command for datatable creation

After defining the field names for table the relations between them have to be defined. This possibility is under Constraints tab where primary and foreign keys can be defined. When defining a relation it's possible to select the desired functioning when key values are changed or records have to be deleted. For example the complete SQL definition to create the Amphibians table is the following (fig. 3).

3.2. Client side

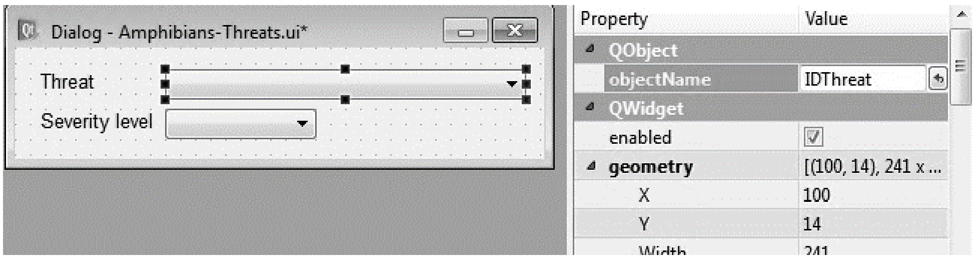
At client side only the QGIS has to be installed. The system is downloadable as standalone application or as part of the OSGeo4W suite with other GIS related applications, libraries etc. As we want to develop our own user interface we downloaded through OSGeo4W which includes Qt4 Designer. Although we have to mention that Qt4 Designer is necessary just to design the user interface, for later use of the project it should not be present on client computers.

The PostgreSQL layers can be added with the Add PostGIS Layers command. At the first use a new PostGIS connection should be defined with proper IP address and database name, also specifying the username and password. Activating the Also list tables with no geometry, all layers will be accessible.

After adding the five layers regarding the amphibians, from which only one has geometry, we can configure that the foreign key values to be turned into intelligible characters. This can be made by selecting the layer properties and at Fields option changing the Edit widget type form Text edit to Value relation (Graser, 2013). For example to have the species names in Amphibians layer instead of their code the following settings has to be made in Value relation window: `Layer - Species | Key column - ID | Value column - name`. Such settings has to be made for IDThreat and severity fields in Amphibians-Threats table also.

With all these by selecting a point which indicates the location of an amphibian we can observe just the species name, without having any information regarding the threats affecting it. To include such an information the effective relation between the Amphibians layer and the Amphibians-Threats layers has to be defined also in QGIS.

One-to-many relations can be defined in the Project Properties option from the Project menu. Six parameters of the relation has to be specified: a user level identification name, the system level identification name, the referencing layer name and the referencing field and the referenced layer and the referenced field. In our case the Referencing layer is Amphibians-Threats while the referenced layer is Amphibians (Kurt et al., 2015).



Property	Value
QObject	
objectName	IDThreat
QWidget	
enabled	✓
geometry	[(100, 14), 241 x ...
X	100
Y	14
Width	241

Fig. 4 - Interface design in Qt4 Designer

Depending on user right the layers can be edited or not, thing valid for all layers whether they geometry or not. The only remaining aspect is related to the user interface. Till now the auto generated interface were used which contains auto arranged elements (often improperly placed) and the label names are taken form field names which in most cases are not so expressive as it should be due to their length limit and space exclusion.

There's a possibility to quickly define user interfaces with Qt4 Designer, part of the OSGeo4W package.

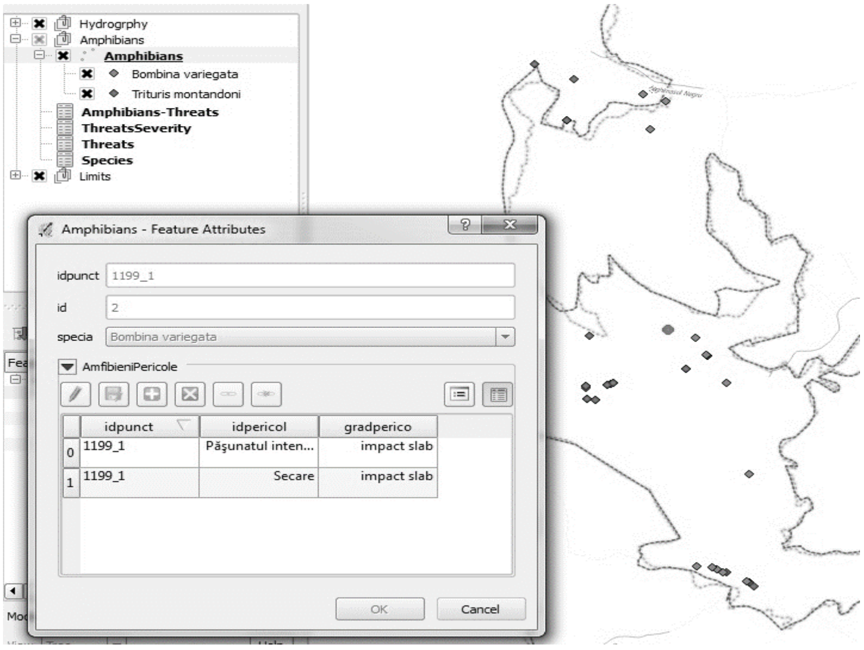
Qt4 Designer gives an intuitive and easy possibility to arrange and label controls. The connection between layer fields and UI controls has to be made by the `objectName` property of the control, which should have the same name as the field name (fig. 4). The resulted and saved file, which has the `ui` extension can be specified for each layer in QGIS from the layer's Properties windows, providing the `ui` file.

Master-slave interface dialogs which can be integrated in QGIS are a little bit harder as it cannot be edited visually. The following code sequence (fig. 5) however integrates the Amphibians-Threats table content into Amphibians UI, so all related threats to a given amphibian is visible in a window, even if the Amphibian layer doesn't contain any information about treats.

```
<widget class="QgsRelationEditorWidget" name="mRelationEditor">
   <property name="geometry">
       <rect>
       <x>20</x>
       <y>40</y>
       <width>651</width>
       <height>261</height>
       </rect>
   </property>
   <property name="qgisRelation" stdset="0">
       <string notr="true">Amphibians-Threats</string>
   </property>
  </widget>
```

Fig. 5. - Design master-slave interface in Qt4 Designer

The `qgisRelation` property holds the relation name defined in QGIS in the Relations section.



idpunct 1199_1
id 2
specia Bombina variegata

	idpunct	idpericol	gradperico
0	1199_1	Păşunatul inten...	impact slab
1	1199_1	Secare	impact slab

Fig. 6. – Client side project with user defined interface working on PostGIS relational database

4. RESULTS

The proposed and developed GIS provides the following novelties/possibilities:

- ➢ over a native desktop GIS:
    - fully relational database, capable to handle primary and foreign keys, maintaining the consistency of the database.
    - user defined user interface including the visualization of layer related data collected by one-to-many relationship
    - user level access control
- ➢ over an on-line GIS
    - desktop GIS type, fully functional data analysis

The resulted GIS software architecture can represent a possibility to combine desktop GIS power with on-line GIS multiuser capability (fig. 6). In such way researchers in various research areas can have a handy and absolutely free tool for a collaborative, yet powerful data storage, management and analysis.

If the desktop component is capable (the majority of the well-known desktop GIS are capable) personalized visualizations can be used to process the same dataset.

R E F E R E N C E S

Chitij, C (2015), *PostgreSQL Cookboo*k, Packt Publishing

Corti, P., Mather, S.V., Kraft, T.J., Park, B. (2014), *PostGIS Cookbook*, Packt Publishing

Schönig, H-J. (2014), *PostgreSQL Administration Essentials*, Packt Publishing

Gerlek, M.P. (2007), *Open Sources*, GEO Connection , Volume 6, Issue 5, pp. 56

Gkatzoflias D., Mellios G., Samaras Z. (2013), *Development of a web GIS application for emissions inventory spatial allocation based on open source software tools*, Computers and Geosciences, Volume 52, pp. 21-33

Graser, A. (2013) *Learning QGIS 2.0*, Packt Publishing

Stone, R., Matthew, N. (2007), *Beginning Databases with PostgreSQL from Novice to Professionals*, Apress

Kurt, M., GISP, Dr. Richard Smith Jr., GISP, Dr. Luigi Pirelli, Dr. John Van Hoesen, (2015), *GISP Mastering QGIS*, Packt Publishing

Lemmens, M. (2008), *Open Source success*, GIM International, Volume 22, Issue 10, pp 8-9

Marquez, A. (2015), *PostGIS Essentials*, Packt Publishing

Nguyen, T.T.T. (2009), *Indexing PostGIS databases and spatial Query performance evaluations*, Volume 5, Issue 3, pp. 1-9