

# The implementation of a relationship database in hidrology. HYDROweb application.

EMIL MARIN POPA, MIHAI VODA, CALIN AVRAM,  
ALINA ELENA PITIC, CRISTINA ELENA RAULEA  
Faculty of Science  
“Lucian Blaga” University of Sibiu  
5-7 Ratiu Street, Sibiu  
“Dimitrie Cantemir” University of Targu Mures  
3-5 Bodon Sandor, Targu Mures  
ROMANIA

*Abstract:* Romania’s river basins need to be monitorized more efficiently for a better water resources management. The collecting data system is based on local operators who measure different parameters at the river basins hydrological stations. They write the measured values on paper every day. At the end of the month they transmit the papers to the Regional Hydrological Institution were other operators introduce the values in the computers. HYDROweb application allows users (local operators) from different hydrological stations around the river basin to introduce directly in the computer the measured hydrological values. Also this program analyses and facilitates the interpretation of the hydrological data.

*Key-Words:* database, templates, web application, river basins

## 1 Introduction

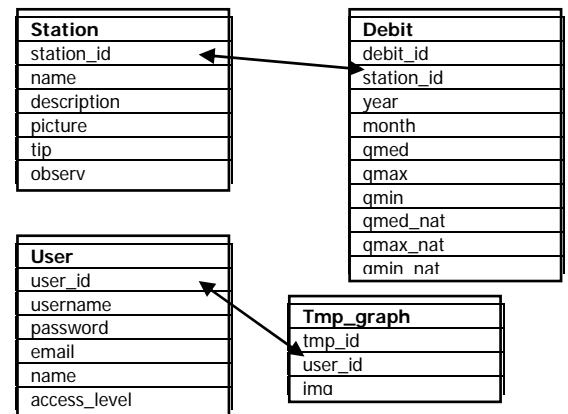
The HYDROweb application allows a potential user to introduce measured hydrological values from many hydrological stations situated in different geographical areas. Also analyses and facilitates the interpretation of the hydrological data.

The application was developed in PHP script and uses MySQL as a database. The application must be installed on a central server with Windows or Linux platform. The operators can access the application directly from the hydrological station using a web browser.

There are two types of users: the system manager and the operators. The system manager creates operators accounts and adds the hydrological stations. The operators can only introduce data and analyses the statistics.

### 1.1 The database

The database is organized in the following tables:



**Fig. 1** – The database general draft

The *Station* table contains information about hydrological stations. The *Debit* table keeps information regarding volumes, and *User* table contains the operators that can access the application.

## 2 Problem Formulation

The application is implemented using the templates philosophy. The templates are html folders that describe the aspect of the web page. The templates contain posted data that are controlled by a PHP script. The templates method allows a fast development of the web application, and also the templates method has a high level of flexibility. The engine used is implemented in the php *Fast Template* class.

The template is an html file that contains some marked blocks overwritten by a PHP script with the useful information.

For example, the next template:

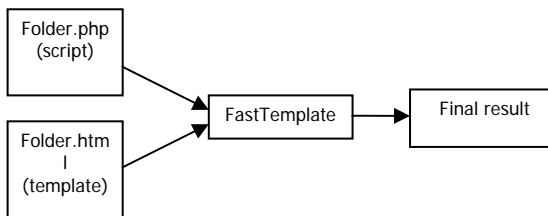
```
<Html>
    <select name="station_id">
        <option
            value="0">Select...</option>
        {STATION}
    </select>
</html>
```

The php script will look as the following example:

```
// is initialed with the folder
$ft = new Fast Template("hidro/tpl/");
$ft->define(array('main' => "sablon.html"));

// is changed by the {STATION} tag with html
code
// necessary for the values definition
$ft->assign("STATION", build_station_list());
$ft->parse('CONTENT', 'main');
$ft->fetch('CONTENT');
```

where the `build_station_list()` function shall generate a string that contains the html code needed for the values definition .



**Fig. 2.** – The templates method general draft

The opening of the each generated page is controlled by a *index.php* folder. The opening of a page is specified through the page parameter and the execution of a specific action is possible by the act parameter. One example could be:

```
http://localhost/index.php?pag=station_list&station_id=8
&act=station-delete
```

The act parameter contains the class name and the function name that will be executed. In this case the interrogation used above will determine the deleting of the hydrological station from the database with id number eight and the posting of remained stations.

In the *func\_perm.php* folder are defined the access permissions for each function. Every operator has an access level defined in the users table. When the function specified in the act parameter is executed the program verifies if the operator has or hasn't the right to execute the specified function.

```
$func_access=array (
    station => array(add=>3,
                    delete=>1,
                    update=>1,
    erase_picture=>1
    )
)
```

The operator with the lowest access level (level one) is considered the strongest and it is the network (system) manager.

Each table has a corresponding class in which adding, modifying and deleting functions have been implemented.

The volumes introduced in the database are interrogated through MySQL interrogations and PHP formulas. The obtained results are then represented using statistical graphs.

The monthly average volumes measured and rebuilt are obtained using the following formulas:

$$\Delta Q = Q_{nat} - Q_{mas}$$

$$\% \Delta Q = \frac{\Delta Q}{Q_{nat}}$$

The implementation of these formulas is possible using MySQL as in the following example:

```
$dbu->query("SELECT station_id,
month, qmed, qmed_nat, (qmed_nat-
qmed) AS dq, (qmed_nat-
qmed)/qmed_nat*100 AS dqp
FROM debit WHERE
station_id='".$globs['station_id']."'
AND year='".$globs['year']."'
ORDER BY month");
```

The result of some interrogation could be seen in the 3rd section. With the obtained data a graph is posted. This is possible using Jpgraph library and the next code:

```
// Create the graph. These two calls are
always required $graph = new
Graph(572,300,"auto");
$graph->SetScale("textlin");
```

```

$graph->SetY2Scale('lin',0,ceil( max($datay_dqp)));
$graph->legend->Pos(0.3,0.9 , "right" , "center");
$graph->legend->SetLayout(LEGEND_HOR);
// Add a drop shadow
//$graph->SetShadow();
// Adjust the margin a bit to make more room for
titles and legend
$graph->img->SetMargin(40,55,40,80);
// Create a bar pot
$plot = new BarPlot($datay);
$plot2= new BarPlot($datay2);
$plot->SetLegend("Qmas");
$plot2->SetLegend("Qnat");
// Adjust fill color
$plot->SetFillColor('blue');
$plot2->SetFillColor('orange');
// Create the grouped bar plot
$gbplot = new
GroupBarPlot(array($plot,$plot2));
// Create a yellow line plot
$plot1 = new LinePlot($datay_dq);
$plot1->SetColor("yellow");
$plot1->SetLegend("DQ[mc/s]");
// Create a red line plot
$plot2 = new LinePlot($datay_dqp);
$plot2->mark->SetType(MARK_UTRIANGLE);
$plot2->SetColor("red");
$plot2->SetLegend("DQ[%]");
$graph->Add($gbplot);
$graph->Add($plot1);
$graph->AddY2($plot2);
// Setup the titles
$graph->title->Set("Variatia debitelor medii
lunare");
$graph->xaxis->title->Set("Luna");
    
```



Login menu for the system manager

**Fig. 3** – Application menu

```

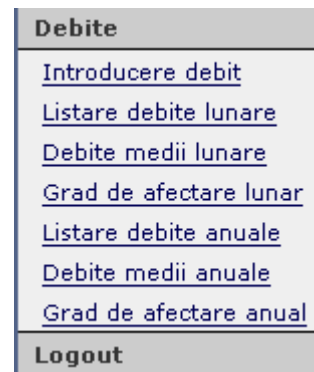
$graph->yaxis->title->Set("Q[mc/s]");
$graph->y2axis->SetTickSide(SIDE_RIGHT);
$graph->y2axis->SetColor('black','blue');
$graph->y2axis->title->Set("Q[%]");
$graph->title->SetFont(FF_FONT1,FS_BOLD);
$graph->yaxis->title->SetFont(FF_FONT1,
FS_BOLD);
$graph->y2axis->title->SetFont(FF_FONT1,
FS_BOLD);
$graph->xaxis->title->SetFont(FF_FONT1,
FS_BOLD);
// Display the graph
$file_name = "grml".rand().".png";
$graph->Stroke($file_name);
$ft->assign("IMG_GRAPH", "<img
src="" . $file_name . \">");
    
```

The result of this code is available in fig 6. The resulted graph is represented with a *png* image format. To avoid problems regarding the browser cache, a random name is used for the image folder that will be different at every generation. This is how the browser will charge each time the correct image for the corresponding graph. The image name will then be stored in the *tmp\_graph* table and will be deleted when operator leaves application.

In the same way is calculated the degree to which the global average volumes is affected but the formula is different:  $k[\%] = Q_{mas} / Q_{nat} \times 100$ .

### 3 3 Problem Solution

*HYDROweb* application displays two types of menus depending on authenticated operator.



Login menu for the operators

**Fig. 4** – Application menu

The system manager can introduce the operators that will use the application. He can also introduce the hydrological stations and data. There is the possibility to introduce a photo representing a geographical map. The

map files are in the upload folder and renamed after *hartă\_id\_statie* form.

Fig. 5 – The volumes table

The application analyses hydrological data listing the measured and natural monthly average volumes and the global affecting level graph.

Luna	Qmas[mc/s]	Qnat[mc/s]	DQ[mc/s]	DQ[%]
Ianuarie	9.09	9.70	0.61	6.29
Februarie	35.20	35.80	0.60	1.68
Martie	43.20	43.60	0.40	0.92
Aprilie	14.60	15.60	1.00	6.41
Mai	14.40	15.40	1.00	6.49
Iunie	23.10	24.20	1.10	4.55
Iulie	14.50	15.50	1.00	6.45
August	11.70	12.80	1.10	8.59
Septembrie	17.10	17.70	0.60	3.39
Octombrie	16.10	16.70	0.60	3.59
Noiembrie	18.90	19.80	0.90	4.55
Decembrie	20.00	20.40	0.40	1.96

Fig. 6 –Measured and natural average monthly volumes

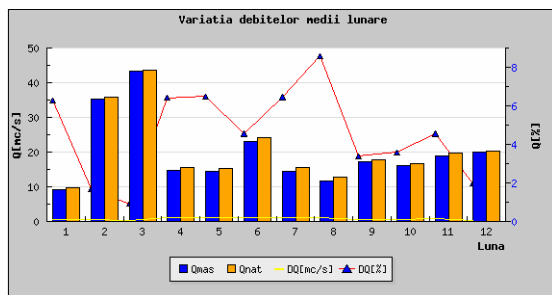


Fig. 7 – Measured and natural average monthly volumes graph

#### 4 Conclusion

The databases have been very useful since they appeared. The main three advantages of using the databases are:

- Centralized data management;
- Data independence;

- Easy integration.

The database management system (DBMS) manages successfully hydrological databases. The access to the hydrological data from a river basin is made through the DBMS very efficiently.

The client/server architecture proved once again its value. Among the advantages of using the client/server architecture we can mention:

- centralizing – access, resources and data security are controlled by server
- updating – the elements can be improved anytime
- flexibility – the new technologies can easily be integrated in the system
- interoperability – all the components (clients, network, servers) work together

The rising popularity of the relational model and the increasing spread of the relational databases are mainly due to their extremely simple but very strong databases software.

Most applications can be accessed via WEB from anywhere in the world. Some of the advantages are:

- Low implementation costs
- Low maintenance costs (hardware, software).

The software such as PHP (Hypertext Preprocessor) or ASP (Active Server Pages) made the via WEB applications highly known. The advantage of using these scripting languages is obvious as all the operations are made on the server. Thus, the client doesn't need a very powerful computer as the server does it all.

#### References:

[1] **DOLLINGER, Robert** –(1998) „Baze de date și gestiunea tranzacțiilor”, Cluj Napoca, Grupul microINFORMATICA

[2] **Gzorodi, Cornelia** -(2000) „Baze de date relaționale. Teorie și aplicații”, Oradea,Editura Treira

[3] **FOTACHE, Marin** – (2001)  
*„SQL. Dialecte DB2, Oracle, Visual FoxPro”*, Iași Editura Polirom

[4] **VODĂ, Mihai** –(2002) *„Metode și tehnici utilizate în evaluarea scurgerii în regim amenajat”*, Cluj-Napoca, Univ. „Babeș-Bolyai”

[4] [www.mysql.com](http://www.mysql.com)

[5] [www.php.net](http://www.php.net)

[6] [www.aditus.nu/jpgraph](http://www.aditus.nu/jpgraph)

[7]  
[www.thewebmasters.net/php/FastTemplate.phtml](http://www.thewebmasters.net/php/FastTemplate.phtml)

[8] [www.phpclasses.org](http://www.phpclasses.org)