**TECHNICAL TRANSACTIONS** | **CZASOPISMO TECHNICZNE**
AUTOMATIC CONTROL | AUTOMATYKA
1-AC/2013

GRZEGORZ NOWAKOWSKI*

# OPEN SOURCE RELATIONAL DATABASES AND THEIR CAPABILITIES IN CONSTRUCTING A WEB-BASED SYSTEM DESIGNED TO SUPPORT THE FUNCTIONING OF A HEALTH CLINIC

# MOŻLIWOŚCI WYKORZYSTANIA RELACYJNYCH BAZ DANYCH *OPEN SOURCE* DO BUDOWY INTERNETOWEGO SYSTEMU WSPIERAJĄCEGO PRACĘ PRZYCHODNI ZDROWIA

Abstract

In this paper the capabilities of using open source relational databases to construct a web-based system designed to support the functioning of a health clinic have been presented as an alternative to commercial solutions. The author introduced a prototype of the system, which is based on selected database. Obtained results confirm the assumption that the medical system does not have to strain health care budget, while providing an acceptable standard of services.

*Keywords: open source, relational database, internet system, health clinic, prototype*

Streszczenie

W niniejszym artykule przedstawiono możliwości wykorzystania relacyjnych baz danych *open source* do budowy internetowego systemu wspierającego pracę przychodni zdrowia jako alternatywy dla komercyjnych rozwiązań. Autor zaprezentował stworzony na podstawie wybranej bazy prototyp systemu. Opisane rezultaty potwierdzają założenie, że system medyczny nie musi nadwyrężać budżetu służby zdrowia, zapewniając jednocześnie akceptowalny poziom świadczonych usług.

*Słowa kluczowe: open source, relacyjna baza danych, system internetowy, przychodnia zdrowia, prototyp*

* Grzegorz Nowakowski, M.Sc., Department of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology.

## 1. Introduction

In recent years the internet technologies have gained immense importance in the management of health care services, not only in case of an individual practitioner, but also at national and regional levels. They offer the possibility to reduce administrative costs and the provision of distance health care in order to avoid an unnecessary duplication of the same medical examinations.

The role of the internet has been constantly growing as it has been widely used by citizens to obtain medical information at present. This situation has drawn attention to the urgent need to elaborate on further information and services in the field of health care and make them available to citizens as well as to ensure the proper quality and safety of internet websites.

Implementation of the system supporting the functioning of a health clinic is usually a very expensive enterprise. Hence, designing a system meant meeting the above-made assumptions, not only the functionality aspects but also the economic option should be considered.

The purpose of this paper is to show the possibility of using open source relational databases to build a web-based system designed to support a health clinic's functioning as an alternative to commercial solutions. For the sake of implementing the assumption, some fully-functional databases systems (including mechanisms such as procedures, functions, triggers and cursors) were tested and compared. Then, the author presented a prototype of the system, which, based on the selected database system, will support health clinics. Obtained results confirm the assumption that the medical system does not have to strain the health care budget, while providing an acceptable level of a given service.

## 2. Open source relational databases review

Open source is a way of creating and distributing software based on a free-sharing system with the source code [1]. The specificity of an open source product distribution allows for an analysis, an independent compilation (creation of the executable option) and modifications to the code by independent developers.

The idea of open source [2] software is gaining more and more importance and attention, and all indications are that we will see its further development in the future. So far, this type of software has been developing in the environment using the internet as a medium. Open source been introduced to the commercial market and changed the prevailing rules of thereof.

Open source software offer covers a wide range of products – from device drivers, usage packages, to server applications and development tools [3].

A relational database is a computer database in which all data is stored in relations which (to the user) are tables with rows and columns [4]. Each table consists of records (called tuples) and each record is identified by a field (attribute) containing a unique value. Every table shares at least one field with another table in 'one to one', 'one to many', or 'many to many' relationships. These relationships allow an database user to access the data in almost an unlimited number of ways, and to combine the tables as building blocks to create complex

and very large databases. It has an internal programming languages, typically using SQL to manipulate the data, by which it creates advanced data services [5].

Relational databases are the basis for most modern systems [6]. Although various database management systems differ from one another in many aspects, they are based on a common theoretical basis.

A brief description of five fully functional (including mechanisms such as procedures, functions, triggers, and cursors) relational open source databases has been presented below.

MySQL is one of the most popular open source relational database management system (RDBMS) [8]. Right now it is being developed by the Oracle company. MySQL is used by some large-scale projects such as Wikipedia, Twitter and Facebook. Oracle Company has optimized MySQL's functioning to work on Windows, so the database system has become fully cross-platform.

MariaDB is a binary equivalent of the MySQL relational database management system [8]. Each month MariaDB code is combined with MySQL code in order to provide support for all the functionalities and add modifications introduced by Oracle. MariaDB is available under GNU license.

PostgreSQL has its roots in the Ingres project, conducted at the University of Berkeley in the 80's of last century. The first version was presented in 1995. It is an open source project based on PostgreSQL license. PostgreSQL is an object-relational database management system (ORDBMS) [10].

Apache Derby is a relational database management system (RDBMS) available as an Apache project [11]. It is written in Java and designed to be used in software written in this language. It has very low system requirements – less than 3 MB of memory, including a built-in JDBC driver. It is available under the Apache license, version 2.0. The database system can operate in two modes: as a full-featured embedded database system or as a traditional database system working in client-server mode using Derby Network Server.

Firebird is an open source SQL relational database management system [12]. The current version of Firebird is based on InterBase, created and developed by Borland in 2000. Developed modules are available on the license of Initial Developer's Public License. Firebird runs under the control of Linux, Unix and Windows [7].

## 3. Comparison of open source relational databases

Discussed databases systems were tested on large data tables. Testing data was a log file (from one of the web server) consisting of a million entries. The same data set was used for all the tested products. Standardized tests such as: mass import of data and basic data operations like SELECT, UPDATE, and DELETE were started. 64-bit Dell servers, running on Windows Server 2008 R2, were used in a test environment. All the tested databases are cross-platform [7].

**Comparison of relational open source databases [7–13]**

| Database | MySQL | MariaDB | PostgreSQL | Apache Derby | Firebird |
|---|---|---|---|---|---|
| **Version** | 5.5.8 | 10.0 | 9.2 | 10.9.1 | 2.5.1 |
| **Management tool** | phpMyAdmin, HeidiSQL | HeidiSQL | pgAdmin | interactive scripting tool called 'ij' | FlameRobin |
| **Dataset import time (one million records)** | time similar to MariaDB | one minute | thirty seconds | few seconds | slower than MySQL, MariaDB and PostgreSQL |
| **Returning time of the first hundred thousand records** | time similar to MariaDB | two seconds | thirty seconds | five seconds | slower than MySQL, MariaDB and PostgreSQL |
| **Performing time of the basic operations (SELECT, UPDATE, DELETE, orders) on smaller datasets** | slower than MariaDB | fractions of a second | half a second | slower than MySQL and MariaDB | slower than MySQL, MariaDB and PostgreSQL |
| **Security** | access control lists for all objects and operations<br><br>SSL communication between the client and server<br><br>cryptographic functions, etc.<br><br>(everything contained in the manual) | access control lists for all objects and operations | authentication through a number of mechanisms from the Kerberosto LDAP<br><br>SSL communication between the client and server<br><br>database can be encrypted with using the pgCrypto extension | authentication and authorizing of users and restrictions on the level of objects<br><br>possibility to encrypt the database | access control at the level of objects and the ability to remove the source code triggers, procedures and views |

| Database | MySQL | MariaDB | PostgreSQL | Apache Derby | Firebird |
|---|---|---|---|---|---|
| **Backup and recovery** | offers a variety of backup and recovery strategies from which we can choose the methods that best suit the requirements for our installation<br><br>(everything contained in the manual ) | backup/ restore tools (mysqldump, xtrabackup, filesystem snapshots, etc) | backup/ restore tools (SQL dump, cluster dump, offline copy backup, continuous archiving, pg_basebackup, point-in-time recovery, pg_upgrade) | provides a way to back up a database while it is either offline or online<br><br>restore a full backup from a specified location<br><br>(everything contained in the manual) | backup/ restore tools (gbak commands, nbackup) |
| **Technical support** | available from Oracle (both free of charge and paid) and many other online sources (for diverse levels) | website design as well as internet forums and blogs | extensive documentation and FAQ, active community of users | apache website contains complete documentation<br><br>active developer community | website contains complete documentation |
| **Additional advantages** | 9 out of 10 websites use MySQL databases (according to Oracle) | mechanism of sub-query (sub-queries) | available in most Linux distributions | low system requirements | possibility of encoding the procedures in Java and C + + (version 3.0) |

As shown in the above comparison MariaDB server is a bit more efficient than the others. Also, it is an open product designed and developed by the same team that created MySOL prior to its acquisition by Oracle. MariaDB can be used by any server that has been used by MySQL up until now.

Table 2

**Summary of tested database servers [7]**

| Database server | Arguments for | Arguments against |
|---|---|---|
| **MySQL** | cross-platform, high performance, a large number of installations, the ability to migrate to commercial versions | some functionalities reserved for commercial versions, the lack of built-in management tools |
| **MariaDB** | performance, cross-platform, commercial independence, completely compatible replacement for MySQL | the lack of built-in management tools |
| **PostgreSQL** | cross-platform, high performance, management tools in the basic installation | sometimes a little intuitive syntax, poor performance of logging |
| **Apache Derby** | cross-platform, low system requirements, can be used as an embedded solution or in a client-server environment | incomplete management tools, trouble with the syntax |
| **Firebird** | cross-platform, easy installation | performance, scalability |

## 4. A prototype of an internet system to support the functioning of a health clinic

Working with an internet system which supports the functioning of a healthcare center logistically corresponds with the order of the stages which the patient goes through in order to benefit from a healthcare center [14]. The system is expected to enhance the doctor's work.

A prototype of the system was developed on the basis of a few monthly interviews conducted with the staff that is employed in Scanmed Multimedis health center in Krakow. This fact enabled the author to design particular modules of the system, and a special emphasis was put on their compatibility with a real activity.

### 4.1. The system concept

Among the users of the system three main groups should be distinguished: patients, doctors and administrators. Access to particular modules of the system is granted for each user on different authorization levels, according to their needs and competencies.

All data in the system are stored on a server. There is information regarding patients, personnel, visits, tests, et cetera there. The common means of communication is the internet.

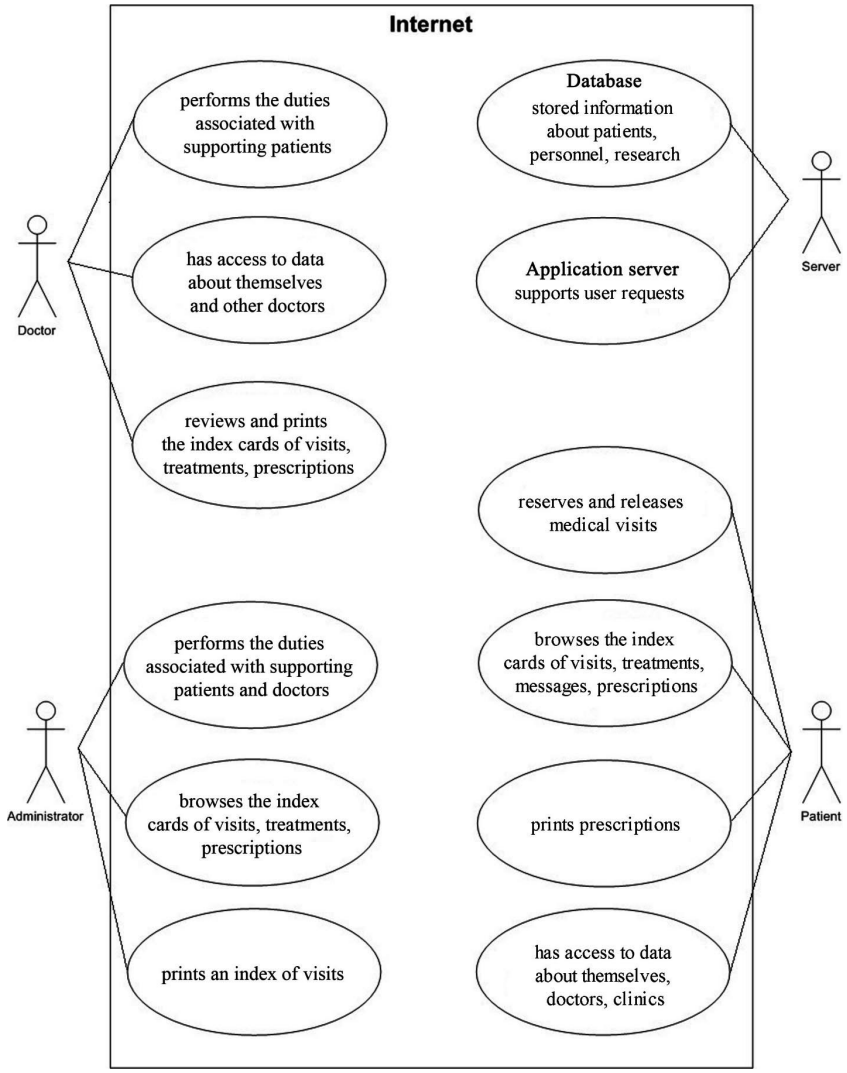


Fig. 1. The system concept

Rys. 1. Koncepcja systemu

## 4.2. System Model Information Flow

The below chart illustrates data processing in each iteration. The starting point depends on the authorization granted based on the assignment to one of the three user groups.

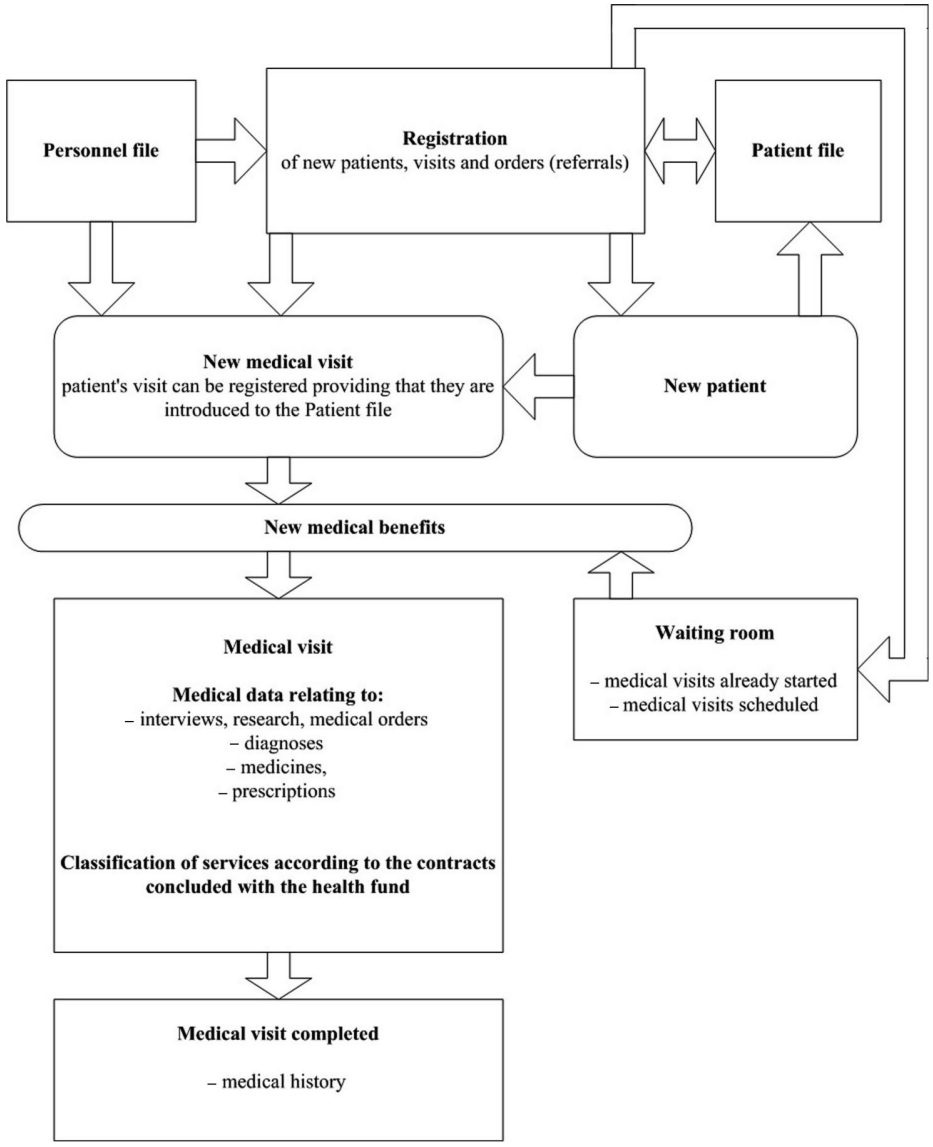


Fig. 2. System Model Information Flow

Rys. 2. Model przepływu informacji w systemie

According to the chart, the system contains the following modules:

- **Personnel and patients file** – for managing information regarding the doctors and the patients of the clinic;
- **Registration** (reservation) – for planning and booking new visits, patients, making accessible the schedule which includes the doctors' allocations and their office hours, daily patients limits;

- **Waiting room** – for browsing through the planned and started visits, ordered by the date and hour of the appointment and by the names of the doctors assigned to carry out the planned visit;
- **New medical visit** – collecting all information regarding the patient's appointments, enabling printing prescriptions;
- **New medical benefits** – collecting all information regarding medical records and list of services according to the contracts concluded with the health fund;
- **Medical visit completed** – for developing medical history records, issuing prescriptions, reporting the registered and elaborated data during the visit.

### 4.3. System Requirements

A proper operation of the system requires the provision of appropriate hardware and network infrastructure. The patient and the doctor find it convenient. It is enough for the system to operate on mid-range computer with access to the internet and mainstream pre--installed software (web browser). A key role of the system is to guarantee adequate storage space of the database and the continuous high-speed internet.

### 4.4. A system architecture

The structure of the system is based on a three-tier architecture [15]. Each of the modules created for the system contains components from one of the three application server layers enumerated below:

- **the presentation tier** consists of the user interface elements of the site, and includes all the logic that manages the interaction between the visitor and the client's business. This tier makes the whole site eye-catching, and the way it is designed is of a crucial importance for the site's success;
- **the business tier** (also called the middle tier) adopts requests from the presentation tier and returns a result to the presentation tier depending on the business logic it contains. Almost any event that happens in the presentation tier usually results in the business tier being called (utilized), except events that can be handled locally by the presentation tier, such as simple input data validation, and so on;
- **the data tier** (sometimes referred to as the database tier) responsible for managing the application's data and sending it to the business tier when requested.

These tiers are purely logical – there is no constraint on the physical location of each tier. In theory, it is possible to place all of the application, and implicitly all of its tiers, on a single server machine, or place each tier on a separate machine if the application permits this [15].

As the below chart shows, most of the data shown to the system users are retrieved from the relational open source database system. As mentioned before, the database is free of charge, fast and reliable. It is also significant, that many companies which provide internet services offer an access to this sort of databases systems, which facilitates launching new websites. This is the reason why this particular platform has been chosen for creating the system prototype.

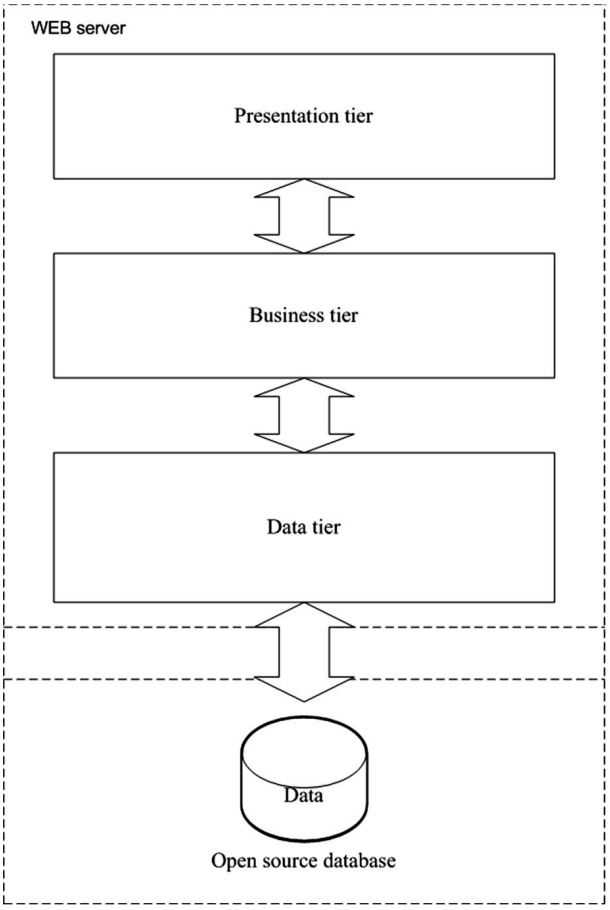


Fig. 3. Open source database server in three-tier architecture

Rys. 3. Serwer bazy danych open source w trójwarstwowej architekturze

It should be mentioned that the important constraint in the three-layered architecture model is that information must flow in sequential order between tiers. The presentation tier may communicate only with the business tier. It is impossible to approach the data tier directly. The business tier constitutes the 'brain' of the application, which communicates with the remaining tiers and processes and coordinates all the information flow.

If the presentation tier directly accessed the data tier, the rules of three-tier architecture programming would be broken [15, 16]. When you implement a three-tier architecture, you must be consistent and obey its rules to reap the benefits.

In order to make the data transfer and processing within separate system tiers understandable, an uncomplicated example will be analyzed: 'The display of a patient's personal details' when the 'My data' button is pressed by the patient. Also it will be indicated which technologies/protocols can be used for a given tier.

When the visitor clicks on the 'My data' button associated with a particular patient (Step 1), the presentation tier (which contains buttons and can be made by means of smarty

componentized templates) forwards the request to the business tier. Information is the order type of 'display personal data of the patient' (Step 2).

The business tier (which can be made in PHP – open source technology for building dynamic, interactive web content) receives the request and interprets it as a command to show the personal data of a particular patient. The service request is forwarded to the command data tier to search for a patient's data (Step 3). The data tier needs to be called because it stores and manages the entire web site's data.

The data tier (in which can be used MySQL) searches for the specified personal information of the patient (Step 4) and eventually returns a success code to the business tier. The business tier (the software instructions to access data (SQL) can be placed in several classes PHP) handles the return code (Step 5) and any errors that might have occurred in the data tier and then returns the output to the presentation tier.

Finally, the presentation tier generates a view of personal information of the patient (Step 6). The results of the execution are wrapped up by generating a HTML web page that is returned to the visitor (Step 7). The personal data of the patient is then displayed in the browser of the patient.

Note that in this simple example, the business tier doesn't do a lot of processing and its business logic isn't very complex. However, if new business rules appear for your application, you would change the business tier. In any case, the presentation tier is informed about the status and provides human-readable feedback to the visitor.

## 4.5. Prospects for the development of the system

A web-based system designed to support a health clinic's functioning should develop very dynamically. The construction of the system's individual parts [17] should change along with the changing legal situation in health care. Continuous cooperation with users (patient, doctor, administrator) should allow for the insertion of facilities for even more efficient operation.

To improve the performance of the existing system the following modules [14] could be added:

- **Statements** – a module would enable the storage of information concerning declarations that has been made (or withdrawn) by patients for primary care physicians within the General Health Insurance. That could control the number of declared patients in the office or clinic and generate reports to the National Health Fund of the declarations made to a particular provider;
- **Medical benefits** – the module would be responsible for the registration of all orders of diagnostic tests, laboratory tests and treatments. Service orders would be controlled from the adoption referrals to the service until the end of the job, which is to be settled with the payer;
- **Refunds** – a module would be responsible for the allocation of costs of the services of the various Departments of the National Health Fund – in accordance with the principles of healthcare financing. The system automatically would generate required reports by the National Health Fund. Thus, a medical unit could be reimbursed for expenses connected with treating patients;
- **Lists** – with increased options for generating different configurations (number and types of services provided, financial statements, cost, etc.) the module allows you to obtain information on the results of the medical unit quickly;

– **Base** – a module would be responsible for managing all databases (dictionaries), e.g. diagnosis, medical services, medicines, dictionary print forms allowing them to define new models. etc. The following dictionary database would be delivered with the system (supplemented partly or as a whole): an address book Branches of the National Health Fund, an address book of the country, the medicine database KS-BLOZ, the diseases database according to X revision of the International Statistical Classification of Diseases and Related Health Problems (ICD-10) – three or four characters and database of medical procedures (International Classification of medical procedures: ICD-9-CM).

### 4.6. A system infrastructure – conclusions

The solution of this infrastructure supports the following arguments:

– availability (each user of the system with an account can use it from the internet);
– possibility to store data (patients, doctors) as well as all documentation (visits, cards, prescriptions) in one place – a relational database open source;
– intuitive user system and thus diminishing the education needs of the system;
– reduction of documentation expenses;
– better organization of work and the use of personnel;
– using the system only when necessary, using it to such an extent that the user is interested in (modularity);
– the PCs and the internet are currently available in each clinic (the result of savings, they do not need to buy new hardware).

## 5. Conclusions

The purpose of this paper was to show the possibility of using an open source relational database system to build a web-based system designed to support health clinic's functioning as an alternative to commercial solutions.

It should be noted that the choice of database system determines the fact how the system will operate in real-world conditions to a large extent. The time of creation of such a system also depends on the database.

The results of standard tests carried out on large tables of data confirmed the assumptions and demonstrated that the database systems of this type are fast, reliable and above all, free of charge. It is also significant that many companies which provide internet services offer access to these databases, which further supports the choice of such a solution.

A prototype system has been designed (by developing a flexible architecture) in such a way that it can be extended with new modules (which can be implemented as separate components) or existing ones may be verified to facilitate the work of a doctor or patient service.

The prototype indicated a relational database open source as a basis system, thus proving the assumption that the medical system does not have to strain the health care budget, while providing an acceptable level of services.

## References


[1] Open Source Initiative, http://opensource.org/ (date of access: 22.06.13).

[2] Gonzales-Barahona J. M., Heras Quiros P., Bollinger T., *A brief history of free software and open source*, IEEE Software, January 1999, 32-33.

[3] Gacek C., Lawrie T., Arief B., *The many meanings of open source*, IEEE Software, vol. 21, 2004, 34-40.

[4] James Hernandez M., *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design (2nd Edition)*, Sybex, 2003.

[5] Whitehorn M., Marklyn B., *Inside Relational Databases 2nd Edition*, Springer, 2001.

[6] Beynon-Davies P., *Database Systems*, Palgrave Macmillan, 2003.

[7] Networld, 2013, http://www.networld.pl/artykuly/387854/ (date of access: 22.06.13).

[8] MySQL http://www.mysql.com/products/, (date of access: 22.06.13).

[9] HeidiSQL http://www.heidisql.com/, (date of access: 22.06.13).

[10] Apache Derby, http://db.apache.org/derby/, (date of access: 22.06.13).

[11] pgADMIN, http://www.pgadmin.org/, (date of access: 22.06.13).

[12] FlameRobin, http://www.flamerobin.org/, (date of access: 22.06.13).

[13] Berkus J., *Ten ways to wreck your database*, O'Reilly Webcast, July 2009.

[14] Pietka E., *An integrated information system for hospital work*, PWN, Warsaw 2004.

[15] Darie C., Balanescu E., *Beginning PHP and MySQL E-Commerce: From Novice to Professional*, *Second Edition*, Apress, 2008.

[16] Software Developer's Journal, 2011, http://sdjournal.pl/magazine/1758-projektowanie-systemow-informatycznych-w-praktyce/, (date of access: 22.06.13).

[17] Coelho F., *Database quality survey projects and results*, January 2012, detailed list of projects surveyed in *On the Quality of Relational Database Schemas in Open Source Software.*