

The development of an electronic business based on the MySQL technology

Andreea IONESCU, Andreea ISTOC

Academy of Economic Studies, Bucharest, Romania

Academy of Economic Studies, Bucharest, Romania

University Hyperion, Bucharest, Romania

Bullguard, Bucharest, Romania

andreea_ionescu_25_2011@yahoo.com

andreea_istoc@yahoo.com

This article aims to demonstrate the importance of using relational database management system in working with web applications. We chose MySQL technology like representative relational database management system because has: portability, scalability, speed, is easy to use, is open source, is widely used by web developers and provides good security.

We watched the analysis of an electronic business. We also presented the technologies that we have used: PHP, MySQL, HTML, CSS.

Keywords: *relational database management system, electronic business, MySQL technology, business to consumer type, security*

1 Introduction

Development of the electronic business is in close contact with good maintenance of databases and with usage of relational databases management system able to do various transactions and to respond in due time needs customers. We observed that for big quantities of data MySQL server provides high performance of insert transactions [1], therefore we have chose to work with this technology.

Development of an electronic business involves ensuring of a good security.

In the future trust and security will lead to electronic business development [2].

To ensure development and improvement of business, e-business uses compliance and communication technologies in large organizations [3].

The current state of knowledge refers to the existence of many types of e-business on the market and the importance of the business models that lead to value creation and performance of the company [4].

In this article we analyze BCOMPUTERS an online business,

which it is based on a MySQL database.

2. The analysis of an electronic business

We chose as case study analysis of the BCOMPUTERS online business, which is based on a MySQL database.

BCOMPUTERS site has the most diversified range of IT products on the Romanian market and it is found in the top of ecommerce sites. BCOMPUTERS is an electronic business to consumer which has: product catalog, a description of these products, products offered in promotions, shopping cart, online secure payment methods (Paypal).

The user is greeted with a friendly interface if he accesses the website. Since it can begin searching for a particular product, he will find products grouped by category. In the middle of the main page are presented products that are on promotions. On the left side of the web application are presented to customer, products that are grouped in categories and subcategories.



Figure 1. BCOMPUTERS electronic business

If the user chooses a category, there is a listing of all products in that category and the user has the option to filter the list of products even more, by choosing a subcategory. When a product attracts the attention of the buyer, he will access the details of that product:



Figure 2. Product's description

When the user accesses the search option from the menu, it appears a list of producers, if the client has an affinity for certain brands. He can choose from the listed brands.



Figure 3. List of Producers

When the user accesses information about a particular product, there are shown the product name, category from which it belongs, the manufacturer, the price and the warranty. In addition, under the product, its technical details are displayed that gives customers information that can

help in the decision to buy or not that product. When a user will be authenticated, he will introduce the username and password:



Figure 4. The Authentication based on username and password

The client uses the shopping cart to buy the products he wants:



Figure 5. Shopping cart

Payment of goods is done through three options: Cash, Card or PayPal, the customer can choose the payment's method that he wants:



Figure 6. Payment products

We also use the PHPMyAdmin management interface. The database that we worked is "calculatoare1.sql" and consists 12 tables with the following structure: produse, producatori, comenzi, categorii, cos_de_cumparaturi, contact, detalii_comenzi, disponibilitate, forum, promotii, categorii and users:

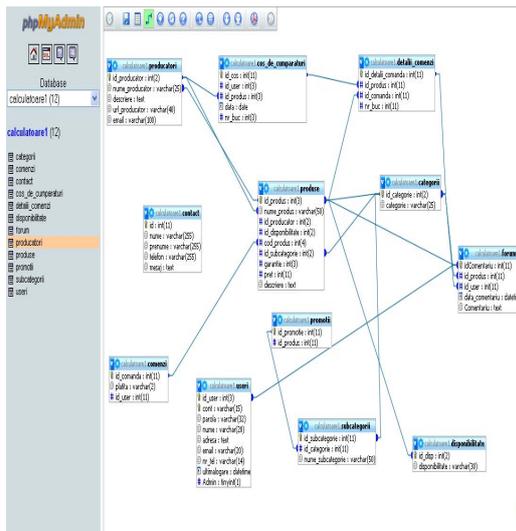


Figure 7. The MySQL database

The Produse table has the following structure:

```

calculatoare1.produse
# id_produs : int(3)
# nume_produs : varchar(50)
# id_producator : int(2)
# id_disponibilitate : int(2)
# cod_produs : int(4)
# id_subcategorie : int(2)
# garantie : int(3)
# pret : int(11)
# descriere : text
    
```

Figure 8. The produse table

The Useri table has the following structure:

```

calculatoare1.useri
# id_user : int(3)
# cont : varchar(15)
# parola : varchar(32)
# nume : varchar(20)
# adresa : text
# email : varchar(20)
# nr_tel : varchar(14)
# ultimalogare : date
# Admin : tinyint(1)
    
```

Figure 9. The useri table

Security of data refers to ensuring the protection and confidentiality of data [5]. The Data Security is ensured by using MD5 algorithm (Message Five Digest). MD5 encrypts the users' passwords in the database.

	id_user	cont	parola	nume
	1	student1	b9acd388cdd72928bc964d81d5858bb7	student

Figure 10. The encrypted password

The cos_de_cumparaturi table has the following structure:

```

calculatoare1.cos_de_cumparaturi
# id_cos : int(11)
# id_user : int(3)
# id_produs : int(3)
# data : date
# nr_buc : int(3)
    
```

Figure 11. The cos_de_cumparaturi table

The comenzi table has the the following structure:

```

calculatoare1.comenzi
# id_comanda : int(11)
# platita : varchar(2)
# id_user : int(11)
    
```

Figure 12. The comenzi table

The categorii table has the following structure:

```

calculatoare1.categorii
# id_categorie : int(2)
# categorie : varchar(25)
    
```

Figure 13. The categorii table

The Detalii_comenzi has the following structure:

```

calculatoare1.detalii_comenzi
# id_detalii_comanda : int(11)
# id_produs : int(11)
# id_comanda : int(11)
# nr_buc : int(11)
    
```

Figure 14. The detalii_comenzi table

The producatori table has the following structure:

```

calculatoare1.producatori
id_producator : int(2)
nume_producator : varchar(25)
descriere : text
url_producator : varchar(40)
email : varchar(100)

```

Figure 15. The producatori table

The disponibilitate table has the following structure:

```

calculatoare1.disponibilitate
id_disp : int(2)
disponibilitate : varchar(30)

```

Figure 16. The disponibilitate table

The forum table has the following structure:

```

calculatoare1.forum
idComentariu : int(11)
# id_producator : int(11)
# id_user : int(11)
data_comentariu : date
Comentariu : text

```

Figure 17. The forum table

The subcategorii table has the following structure:

```

calculatoare1.subcategorii
id_subcategorii : int(11)
# id_categorie : int(11)
nume_subcategorii : varchar(50)

```

Figure 18. The subcategorii table

The contact table has the following structure:

```

calculatoare1.contact
id : int(11)
nume : varchar(255)
prenume : varchar(255)
telefon : varchar(255)
mesaj : text

```

Figure 19. The contact table

3. Technologies used:

To achieve this electronic business we used the XAMPP platform, that contains the Web Apache server, the MySQL database server, PHP and PERL.

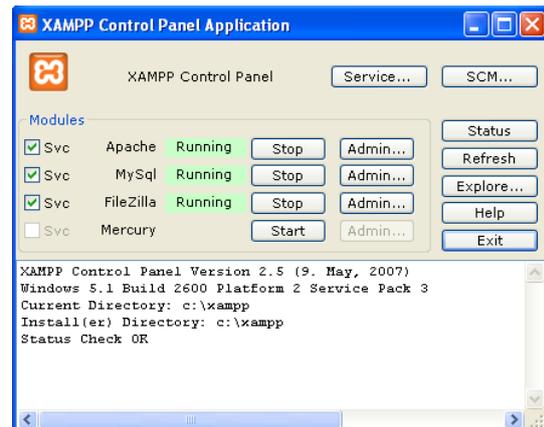


Figure 20. The XAMPP Control Panel

The XAMPP platform is easy to install and configure and it is implemented on a variety of operating systems such as Linux, Windows, Mac OS X, Solaris. For the development of BCOMPUTERS electronic business which is located at address: <http://bdcomputers.twostudio.ro> on the Internet, we used the following technologies: MySQL, PHP, HTML, CSS.

PHP (Hypertext Preprocessor) is one of the most popular programming languages who is used for development of web applications. He runs in server mode and is an open source product distributed. Currently reached version is 5.3.6.

The MySQL technology is an relational database management system.

We chose this technology because has many advantages: it is easy to use, can work with large databases. The current version is 5.0.8.

We used css styles to format the web pages:

```

<style type="text/css">
ul

```

```

{
align:center;
width:0;
padding:0;
margin:0;
style:Tahoma;
border-color:#000000;
}
a
{align:center;
style=Tahoma;
float:left;
width:140;
text-decoration:none;
color:black;
background-color:#3366CC;
border-color: 10 px solid black;
border-color: #000000;
}
a:hover {background-color:#3366CC;}
li {display:inline;
}
</style>

```

4. The main commands in SQL

SQL has the first implementation in 1970 and it is based on the studies of E. F.Codd. SQL is an standardization language and it can be used to access databases: Oracle, SQL Server, DB2, MySQL.

SQL commands are grouped into five categories as follows:

- a) The query language
- b) The data manipulation language
- c) The data definition language
- d) The transactions control language
- e) The data control language

The query language allows retrieving of lines stored in database tables. The SELECT command is used. In SQL the query data stored in databases is done with the SELECT command. The SELECT command implements all operators of relational algebra. The SELECT command syntax is:

*SELECT attribute
FROM object*

After SELECT keyword is specified list of attributes to be returned as a result of the query and after keyword FROM are specified objects (tables,

views, synonyms) from which we can select those attributes.

a) The data manipulation language allows you modifying the content of the tables. The used commands are:

-the INSERT command is used to add new lines in a table.

Also the INSERT command is used for adding new rows in a table or in tables previously created as basic views.

The INSERT command can be used in two ways:

b.1.) To enter data into a table, one record at a time. In this case the syntax is:

```
INSERT INTO table [(column1,
column2)] VALUES (value 1, value 2);
```

b.2.) To enter data into a table by copying many records in another table or group of tables, these recordings are the result of the SELECT command, in this case the syntax is:

```
INSERT INTO TABLE [(COLUMN 1,
COLUMN2, ...)] SELECT_COMMAND;
```

The UPDATE command- it is used to update data stored in a table.

The UPDATE command is used to modify existing values in a table or in a basic table and has the following syntax:

```
UPDATE TABLE [ALIAS]
SET award_columns, [award_columns,
...]
```

[WHERE CONDITION];

WHERE award column can have the one of forms:

-column= {(expression)|(subquery))
or

(column [,column]...)=(subquery);

There are two ways for updating:

->providing explicitly for each value or expression to be modified;

-> obtaining values with a subquery;

The update command changes the values of records depending on the condition of where clause. In the given table in the absence of where clauses all of records will be updated.

DELETE command – it is used to delete lines in a table.

The DELETE command performs deleting records from a table or tables of a view depending of a certain condition and has the following general syntax:

```
DELETE FROM TABLE [WHERE
CONDITION]
```

The DELETE command delete some records depending on the condition of where clause. Without the where clause will be deleted all entries in the table given. In this clause may be included and subquery.

c) Data definition language allows the definition of a structure of tables that compose databases. The commands that we use to implement data definition language are:

CREATE command- for creating a database structure or a table.

In the first case for creating a database structure the syntax of CREATE command is:

```
CREATE          DATABASE
DATABASE_NAME;
```

For creating a table of a database the syntax of CREATE command is:

```
CREATE TABLE "table_name"
("column1"
```

```
"data_type_for_column_1",
```

```
"column2"
```

```
"data_type_for_column_2", ...)
```

The ALTER command for changing the structure of a database or a table.

The ALTER TABLE command:

-add a column;

-modify an existing column;

-defines a default value for a column;

The syntax of ALTER TABLE command is:

```
SQL>ALTER TABLE
      ADD      (column      datatype
[DEFAULT expression]
[, column datatype]);
```

```
SQL>ALTER TABLE table
MODIFY (column datatype [DEFAULT
expression]
      (, column datatype));
```

The ALTER TABLE is useful if you want to change a table structure. You can add new columns using the ADD clause:

The syntax is:

->table->table name;

->column->the name of new column;

->datatype-> the type date and length;

-Default expression->the specific default value for column;

You can modify an existing column using modify clause.

The DROP command- is used to delete the structure of a database.

The instruction SQL DROP TABLE is the SQL command that removes a table from a database.

The SQL DROP TABLE syntax is:

```
DROP <table_name> (<column_list>)
```

The RENAME command-is used to change the name of a table.

d) To implement the tranzactions control language we use the following commands:

-The changes made to database to become permanent we will use the COMMIT COMMAND;

-The ROLLBACK command-possible to dispense with the latest changes to the databases;

-The SAVEPOINT command –defines a rescue at which it can return and gives the changes made after this point to the database.

e)The Data control language allows the definition and modification rights on databases. The used commands are:

The GRANT command – for granting rights to other users of the database.

The REVOKE command – to cancel the rights of certain users.

5. The main functions

In tables from queries and reports we can define fields that contain results of calculations with data values in database

tables. These results come from the evaluation of arithmetic expressions that contain: numbers, names of table columns (attributes) and functions of the standard SQL language. Best known arithmetic operators for real numbers are: +, -, *, / and for integer numbers are: +, -, \, mod.

The functions are classified in two types:

a) The functions on a several records-functions that calculate a value based on the values of a column of the table (depending on aggregate):

-The AVG() function - returns the average;

-The COUNT() function-returns the number of rows;

-The MAX() function -returns the maximum value;

-The MIN() function-returns the minimum value;

-The SUM() function -returns the sum of the values

-The VAR function-returns dispersion of the values from the respective column;

- The FIRST() function- returns the first value;

-The LAST() function- returns the last value;

b) The functions that works with the strings:

-The LCASE() function -converts a string to lowercase;

-The UCASE() function - converts a string to uppercase;

-The MID() function—extracts a substring from a string;

-The LEN() function –gives the length of a string;

-The FORMAT() function-formats a field;

-The CONCAT() function - is equivalent with the concatenation operator;

-The ROUND() function- rounds a numeric field to a specified number of

decimal places;

-The NOW() function-provides a string with current date and time;

-The format of the function is:

->format (column_name, format);

-where format is a format string that gives the value of the column. For example the function:

->format (now(), ‘YYYY-MM-DD’)

gives the current date as year-month-day;

The MID() function has prototype:
-mid(column_name, index, [, length])

where:
-column_name is the field which is extracted substring;

-index indicates position of the first character in the subsequence string (first character is string has index1);

-the optional length parameter gives the length of substring. If this parameter is not specified, the function gives the rest of the string.

The Round() function is:

->round(column_name,
number_of_decimal_places)

->the function rounds to the nearest whole number;

The remaining functions have like argument the column name (attribute) of the table.

6. The E. F. Codd's rules

E. F. Codd (IBM researcher) has formulated 13 rules that express the maximum requirements for a database management system to be relational. The rules are useful for evaluating performance of relational database management system.

R0.The relationship-level data management languages used to operate with relationships (information unit).

R1. The logical representation of data: all information in the database must be stored and processed as tables.

R2. Guaranteeing access to data: the

data manipulation language must allow access to each atomic value in the database (table, column, key)

R3. The NULL value- declaration and processing must allow null values as missing or inapplicable data.

R4. The Metadata-information describing the database is stored in the dictionary and treated as tables, as well as actual data.

R5. The used languages- the databases management systems must allow the use of several languages, including at least one allowing define integrity constraints, data handling, access authorization, transaction processing.

R6. The updating the virtual tables- should be allowed as virtual tables and actually be updated (e.g. attribute "value" of a virtual table cannot be updated).

R7. Updates to the database- handling of tables must be done by retrieving and updating operations.

R8. The physical data independence, change the physical structure of data (mode of representation (organization) and how to access) does not affect programs.

R9. The logical data independence, changing the structure of data (logical) tables does not affect programs.

R10. The integrity constraints- they must be defined by data definition language and stored in the database dictionary.

R11. The geographical distribution of data: the data manipulation language must allow application programs to be the same for both distributed data and

for pooled data (the database management system will be in charge with data allocation and with location).

R12. Data processing at low level- If the database management system possess a low level language (the data is processed at registration), it should not be used to avoid restrictions of integrity.

7. The MySQL's characteristics

MySQL as we have stated is a management system relational database and implements SQL (Structured Query Language), which is the standard query language of relational databases. We choose MySQL because it is open source, is the most widely used and can handle very large databases.

Table 1. MySQL's characteristics

MySQL's characteristics	
<i>1.High Performance</i>	MySQL can run on a variety of operating systems such as: Linux, Windows, Mac OS X, Solaris. Meanwhile, he can work with millions of tranzactions.
<i>2.Easy to use</i>	MySQL is easy to conFigure and administered and has a high performance.
<i>3.Scalability</i>	MySQL is open source and it can be easily adapted by the user for his own requirements. It also manages very large databases.
<i>4.Speed</i>	Implements a variety of customer interfaces such as: Java, PHP, Python, C++, C, Perl and provides a great

	data processing speed.
5. Good security	MySQL provides increased security through the implementation of encrypted passwords and ensure data protection through specific mechanisms.

MySQL has a high performance. He can run on a variety of operating systems such as: Linux, Windows, Mac OS X, Solaris. Meanwhile, he can work with millions of transactions.

MySQL is easy to configure and administered and it has a high performance.

MySQL is open source and it can be easily adapted by the user for his own requirements. It also manages very large databases.

Also it implements a variety of customer interfaces such as: Java, PHP, Python, C++, C, Perl and provides a great data processing speed. MySQL provides increased security through the implementation of encrypted passwords and ensure data protection through specific mechanisms.

8. Conclusions

We choose working with MySQL technology because has many advantages: high performance, easy to use, scalability, speed, increased security.

References:

[1]. A. O. Afolabi, A. O. Ajayi, "Performance Evaluation of Database Management System (A case study of INTERBASE and MySQL)", *Journal of Engineering and Applied Sciences* 3 (2), Medwell Journals, 2008, ISSN: 1816-949X, pp. 155-160;

[2]. K. Changsu, T. Wang, S. Namchul, K. Ki-Soo, "An empirical study of customers' perceptions of security and trust in e-payment systems", *Electronic Commerce Research and Applications* 9 (2010), pp. 84-95;

[3]. K.E. Corey, M.I. Wilson, "e-Business and e-Commerce", *International Encyclopedia of Human Geography*, Elsevier, Oxford, 2009, ISBN 978-0-08-044910-4, pp. 285-290;

[4]. K. Wikstrom, K. Artto, J. Kujala, J. Soderlund, "Business models in project business", *International Journal of Project Management*, Volume 28, Issue 8, IRNOP 2009 in Berlin, December 2010, ISSN 0263-7863, pp. 832-841;

[5]. M. Turle, "Data security: Past, present and future", *Computer Law & Security Report*, Vol. 25, Issue 1, 2009, ISSN 0267-3649, pp. 51-58;



Andreea IONESCU graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Academy of Economic Studies in 2008(Bachelor's degree) and Master of E-business in 2010. She is university assistant in computer science at The Faculty of Economic Sciences at Hyperion University of Bucharest. She is currently a PhD candidate at Institute of Doctoral Schools at Academy of Economic Studies from Bucharest, in Economic Informatics Field. Her interest includes: e-commerce and project management.