

Z6. GUI tools for database used in Customer Relationship Management (CRM) system

During the implementation of previously performed laboratory tasks, the main tool for managing the MySQL engine was web-based tool - phpMyAdmin. Performing administrative tasks from this program has many advantages:

- does not require installing any client program on a local machine of the developer,
- is fully mobile, does not require defining communication parameters between the database (DB) server and the client program,
- it allows you to work remotely using the web browser installed on your computer and also on your tablet or smartphone,
- it works well during the implementation of the simple applications using DB, in which there are no complicated relations in DB structure.

However, DB management using a web browser for complex projects is usually not convenient. In such situations, dedicated graphic tools for DB management (GUI Tools), installed on Windows, Linux or MacOSX operating systems, works better. Some tools are designed exclusively for MySQL, others such as HeidiSQL allow you to manage several different DB engines like: MySQL, PostgreSQL and MS SQL. Below there is a list of popular free applications or applications that have at least free limited versions, that allow you to manage MySQL engine.

Free tools

MySQL Workbench	http://dev.mysql.com/downloads/workbench
HeidiSQL	http://www.heidisql.com
Sequel Pro	http://www.sequelpro.com
MyDB Studio	http://www.mydb-studio.com
DBeaver	https://dbeaver.io
DbVisualizer	https://www.dbvis.com
Toad for Windows	https://www.quest.com/products/toad-edge

Commercial tools, with free versions limited for certain time or for certain applications

Navicat for MySQL	https://www.navicat.com/en/products/navicat-for-mysql
SQLyog / SQLMASTERO	https://www.webyog.com/product/sqlyog https://www.sqlmaestro.com/products/mysql
SQLWave	http://www.sqlwave.com
dbForge Studio	https://www.devart.com/dbforge/mysql/studio
DBTools Manager	http://www.dbtools.com.br/EN/dbmanagerpro

Professional GUI tools for DB management require large financial outlays. Development and maintenance of them for many years is not possible only thanks to the free work of developers, testers and volunteers. Therefore, some of the above projects change or will soon change the financing model. Currently free applications may be payable in the future and vice versa. Some projects will also fail if their users would not support them financially. Several known tools of this type are no longer being developed and their existing users had to switch to other programs. That is why is so important to make the right choice of the main GUI tool for DB, which we want to use for a long time and we get to know all its functionalities. If for some reasons we have to change this tool, it can cause:

- the need to choose new tools, which in large corporations cooperating with many groups of programmers is an expensive and sometimes long-lasting process,
- a temporary decrease of the effectiveness of DB developers and architects,
- more frequent errors in the code being created or the code can become less effective for some time.

The choice of the basic GUI tool for creating and managing DB should be (in my opinion) based primarily on:

- the most frequently used and highest rated tools for managing a given DB engine, in our case, MySQL
 - to investigate that, you can generate a question on Google.com: "Best MySQL GUI Tools",
 - however, you have to remember that the creators of some of the "moderated" rankings are companies that occupy top positions in them and you can't rely on all rankings,
- universal tools that allow management of various DB engines, e.g. MySQL, PostgreSQL, Microsoft SQL Server, mastering such tools strengthens our code creation flexibility, which can be based in that case on various DB engines,
- tools that have a stable source of financing - large corporations are behind them (you need to do quite time-consuming research on the network to see who finances particular tools),

- tools that are used by large corporations, for which a frequent change of GUI tools for DB is a very large cost and a temporary decrease in the efficiency and reliability of the generated software (you need to do quite time-consuming research on the web to see which tools are used by the largest corporations).
1. Please review the above projects related to DB GUI tools and select the most promising project in your opinion or project you have used before, in order to implement the following laboratory task. If you do not have time to form your private opinion about it - use MySQL Workbench.
 2. Create an online CRM (Customer Relationship Management) application defined as:
 - a. a set of tools and procedures important in managing customer relations,
 - b. an application used in companies where constant contact and satisfaction with it is a key business value,
 - c. an application enabling uniform access to information about the client in all business processes - starting from the process of selling services / goods, through the current service support, selling subsequent services / goods,
 - d. a set of tools for the generation of relevant statistics used to create a portfolio of services / products.
 3. There is only time in our lab to create a very limited CRM system, with minimum features that are shown below
 - a. Customer registration in the portal:
 - i. entering the login once and entering the password twice in order to enter and verify the identity of both entered forms of the password,
 - ii. the letters of the password should be displayed as asterisks '****' (input type = "password"),
 - iii. logging should call the CAPTCHA function (<https://en.wikipedia.org/wiki/CAPTCHA>) or should call the reCAPTCHA function (<https://en.wikipedia.org/wiki/ReCAPTCHA>) to prevent the automatic machines from creating accounts (web robots) - in this particular case, please **CANCEL** this because it extends the process of checking this exercise by the lecturer,
 - iv. logging should contain mechanisms that enforce the minimum password length, e.g. at least 8 characters, and mandatory characters in the password, e.g. at least one capital letter, at least one digit, etc. - in this case, please **CANCEL** this because it extends the process of checking the task by the lecturer,
 - b. Logging in to the portal with the previously registered login / password,
 - i. The fact of logging in of the client should be registered in DB: date and time, customer name, client IP, client browser type and client operating system, the type of client browser and the client's operating system are data that can be obtained using the links below
https://www.w3schools.com/php/func_misc_get_browser.asp
<http://php.net/manual/en/function.get-browser.php>
 information about the types of browsers and client operating systems are needed to optimize the CRM system for browsers that customers use.
 - ii. Choosing by the client from the combo list one of the predefined issues - the initial division of issues into groups allows employees to specialize in specific issues, e.g.
 - Sales of new services,
 - Technical Support,
 - Cancellation of the service,
 - Other.
 - iii. Asking by the client questions that will be stored in DB along with the previously selected issue.
 - iv. When the employee responds to a given customer's question, then the client will be able to
 - see the question you have asked and the answer given by the employee, the name of the employee,
 - assess the quality of service by entering the rating on the scale from 1 to 5.
 - c. Logging in to the portal an employee
 - i. Choosing by the employee the issue in which a given employee specializes - then only the questions assigned to the above categories are displayed: Sales of new services, Technical support, Cancellation from service, Other.
 - ii. Selecting a customer's question that the employee wants to answer
 - iii. The answer to this question and its registration in DB.
 - d. Login to the portal the staff manager - login / password: admin / admin, this account is needed to assess the number and efficiency of customer service employees and verify the ratings that employees receive from customers,
 - i. Display the amount of all queries generated by all clients.

- ii. Display the amount of all responses given by all employees.
 - iii. Displaying the amount of all responses given by all employees, grouped by employee (clue grouping by employees: https://www.w3schools.com/sql/sql_groupby.asp)
 - iv. Displaying the amount of all responses given by all employees with grouping by issues (grouping tips by: https://www.w3schools.com/sql/sql_groupby.asp)
 - v. Displaying the average rating of all employees assigned to them by clients, grouping them into employees (clue grouping by: https://www.w3schools.com/sql/sql_groupby.asp)
4. Each laboratory group should build a CRM system for a specific IT industry:
 - a. Provision of Internet access services,
 - b. Provision of mobile telephony services,
 - c. Provision of motor insurance OC, AC, and NW.
 - d. Remote management of servers and services provided on these servers,
 5. Create several clients and several employees and test if the dialogue between the service and customers is organized correctly.
 6. Expand the system allowing the client not only to generate a question and get a response from the service staff, but also to ask the employee another question in relation to the received response and get another answer - just like in text messengers - to conduct a dialogue.
 7. Report completion of the task.

Tables (proposals)

workers (idworkers, name, password)	// worker1/pass1, worker2/pass2, worker3/pass3
clients (idclients, name, password)	// client1/pass1, client2/pass2, client3/pass3
logs_workers (idlogs_workers, idw, datetime)	
logs_clients (idlogs_clients, idc, datetime, web_browser, os_system)	
items (iditems, name) - a table with issues allowing pre-selection of the reported problem	
posts (idposts, idc, idw, idi, datetime, post_client, post_worker, rank) – a table with the dialog between workers and clients	