

CMS Query Suite

CS4440
Project Proposal

Chris Baker
Michael Cook
Soumo Gorai

I) Motivation

Relational databases are great places to efficiently store large amounts of information. However, information that cannot be accessed is useless; there must be an interface to retrieve information. If the information is designed to be only retrieved and used by a program, then SQL code and stored procedures provide ample access. There are also scenarios where the information needs to be retrieved directly by users who are trying to find a specific piece of information. It is not reasonable to expect end users, who may not have any SQL experience, to write SQL code to query a database. To improve the SQL experience, there can be enhanced data visualization, statistical analysis, query optimization, and defense mechanisms against SQL injection attacks. In order to avoid having to build a custom interface for querying a new database or finding statistical information about the database, it would be nice to be able to simply configure a generic tool for any database system.

MySQL is a popular relational database management system (RDBMS) among those who are looking to conserve resources because of its good performance and low cost. For example, a startup company may decide to use MySQL because it does not yet have the resources to spend on a more expensive RDBMS. For a company that is just starting, it is important to be able to focus on their main product and market it. With that concept in mind, an end user query interface can be configured to work with any MySQL database. Those who choose to use this product will be able to save the time and money that would be spent developing their own querying interface.

Some concerns with any software product is how many platforms it can run on without having to customize the code too much and what is required of users to run the software. Web applications are naturally cross-platform compatible and require only a web browser on the client side. Since the server side can be much more easily controlled, cross-platform ability on the server side is less of a concern. However, for maximum flexibility and usability, a web application that is to be distributed to many clients needs to be cross-platform compatible on the server side as well. If written in PHP, the application can run on most web servers for easily accessible cross-platform access.

II) Objectives

- To create an end user query interface called CMS Query Suite (CMS) that can be configured to work with any MySQL database.
- To have CMS to be able to adapt to the custom needs of the client and provide a flexible interface for its users through custom configuration.
- To write CMS as a PHP web application, so the software can be run from most web servers for easily accessible cross-platform access.
- To allow multiple users to query the same database simultaneously through a web-based application.

III) Related Work

There are many query building interfaces available at the commercial level, such as Business Objects, Active Query Builder, Open QueryBuilder, FastQueryBuilder, FlySpeed SQL Query, and EasyQuery.NET (easily found using Google). These solutions interface with multiple database platforms, and most of them offer user-friendly drag-and-drop functionality. Currently, however, there are no open source applications for easy query-building for the MySQL platform. The current applications are only useful for knowledgeable users, such as DBAs, but are not suited for general users who are not interested in learning specifics of detailed queries or dealing with schemas and relations. These applications also tend to be stand-alone applications that run on a single machine. Our web-based application will allow multiple users to query the same database simultaneously. Finally, much work has been done to enhance the MySQL query features for PHP version 5, which will be the base of our project and will expedite the application's development.

IV) Proposed Work

Design & Implementation

We will create a general database tool that can be used by anybody to access the information held in it. More specifically, we will create a graphical user interface (GUI) that will allow the user to enter the contents of a search query. The GUI contents will be formatted based on a configuration file and turned into an SQL query in the background and sent to the database. We will make sure the code sent by PHP has been checked for possible SQL injection attacks. The database will execute the query and return the search results. The results will be displayed in the Search Results window. The GUI will be created using the PHP programming language and be displayed in a web format. Before the search GUI is executed, the client should create the database and then specify the queries and their mapping to database objects using the configuration file. The search query will be optimized for efficiency before being sent to the database.

Evaluation of Success

If the time taken by the GUI search approximately matches that of the SQL command-line method, then the GUI system can be considered acceptable. Also, the learning rate of the user to get accustomed to the GUI will also be considered. If the learning rate of the advanced features is low (approximately 3-5 trials), then the GUI can be accepted as appropriate software. Of course, if we decide to publish it and greater the number of downloads, then greater is the success of this project. If the query optimization reduces the performance time of the GUI compared to that of the un-optimized SQL command line queries, then this project would look promising. Also, for the complex search queries, if the actual search results match the expected result, then the software would be a considerable success.

New Ideas and Techniques

The new idea is that this is a database tool that any institution can use to setup a search query interface. The company's desired query types would be set through a configuration file that would set up the database as per the requirements of the institution. Furthermore, queries are not limited to just one table. Data can be extracted from more than one table and packaged together to completely satisfy the end user request. Another novel technique is query optimization. Unlike in raw MySQL queries, query optimization potentially speeds up processing time.

Architectural Design

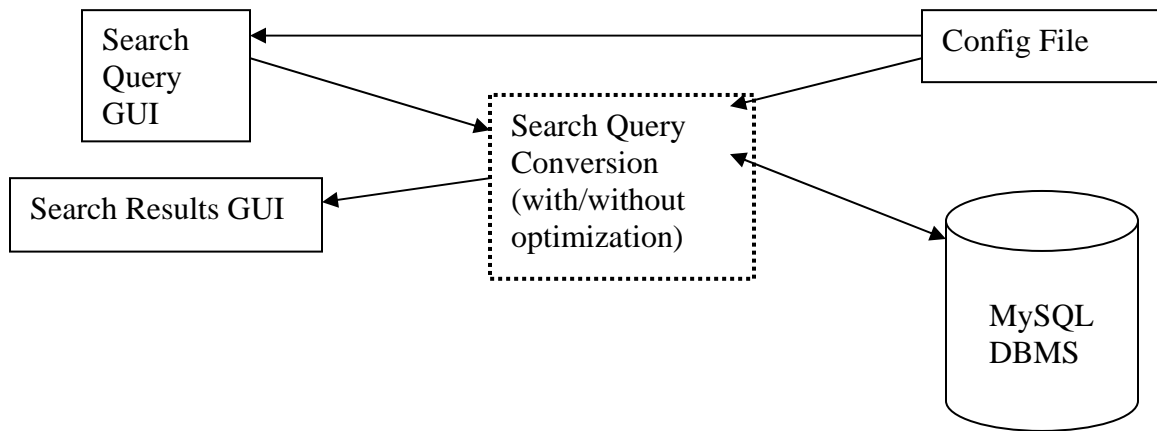


Fig. 1 Architectural Design

As Fig. 1 indicates, the components consist of a MySQL Database Management System (DBMS), Search Query and Results GUIs, and the Configuration(Config) File. The GUI contents are displayed according to the setup defined in the Configuration File. The GUI information is converted into an SQL query using the format mentioned in the Configuration File and then sent to the DBMS. The DBMS sends the appropriate results back to the Search Query process that relays the appropriate information back to the Search Results GUI to be displayed to the user.

V) Plan of action (resources, schedule, and plan for evaluation)

- Resources (software, hardware, platform considerations)
 - Michael's Computer Specifications
 - Linux Computer:
 - 1 GB RAM
 - SUSE Linux 10.1
 - Pentium D 3.0 GHz
 - MySQL Server 5.0.18-Max
 - PHP 5.1.2
 - Windows Computer:
 - 2 GB RAM

Windows XP Professional Service 1.4.2
 AMD Athlon 64 X2 Dual Core Processor 4600+

- Soumo's Computer Specifications

Windows XP Pro SP2
 1 GB RAM
 Pentium 4 CPU 2.80 GHz
 MySQL Server 5.0.37
 PHP 5.2.4

- Chris's Computer Specifications

Laptop

Windows XP Home SP2
 512MB Ram
 Athlon Turion 64 Mobile 1.6GHz
 Eclipse 3.3.0, with PDT, SVN

Desktop

Windows XP Pro SP2
 1 GB Ram
 Pentium 4 2.8 GHz
 MySQL Server 4.0.20a-nt
 PHP 4.3.9

- Resources Acquisition.

Michael has a server set up on his computer with PHP 5 and MySQL 5. Chris has MySQL 4 running on his computer. Soumo has MySQL 5 running on his computer.

- Weekly schedule/milestones

Week	MileStone	Who	Comments
9/16 – 9/22	Project Group	All	9/20
9/23 – 9/29	Proposal	All	
9/30 – 10/6	Proposal/Design	All	Due 10/2
10/7 - 10/13	Feedback from Professor-TA/Design	Professor Liu/TA Gong Zhang, All	Due 10/11
10/14 – 10/20	Design	All	Due 10/20
10/21 – 10/27	Implementation	All	Start 10/22
10/28 – 11/3	Implementation	All	
11/4 - 11/10	Implementation	All	
11/11 – 11/17	Implementation	All	Due 11/17
11/18 – 11/24	Testing/Debugging	All	Due 11/24
11/25 – 12/1	Final Touch/Project Due	All	FT: 11/25 – 11/26; Due 11/27
12/2 – 12/8	Project Demos	All	12/4 – 12/7 TBA

VI) Evaluation and Testing Method

- Speed: GUI vs. SQL

The time for an experienced database user to perform a query both with our tool and also with plain SQL code will be computed with a timer and compared. We expect there to be significant differences in time (in our favor) for larger queries.

- Usability

A user outside our group will be recruited to use the tool, to make sure it is usable. We expect the user to be able to intuitively perform a query after being told about what is stored in the database.

- Optimization

To evaluate how well our query optimization works, we will perform the same query with our optimization procedures and without. For queries against very large datasets, we expect to see a substantial time difference favoring the optimization.

- Result Validation

The result will be validated by verifying queries return the expected results. The database contents will be queried manually to verify data integrity.

VII) Bibliography

Google Code. Google 2007. [GNU Lesser General Public License](http://code.google.com/p/mysql-query-builder).
<http://code.google.com/p/mysql-query-builder>.

MySQL AB. [MySQL 5.0 Reference Manual](http://dev.mysql.com/doc/refman/5.0/en/tutorial.html). 2007.
<http://dev.mysql.com/doc/refman/5.0/en/tutorial.html>

The PHP Group. PHP: A simple tutorial. 2007. <http://us.php.net/tut.php>