Web Secure

PO Box 2726 Cairns, Qld 4870 Australia

P: 1300 73 83 01 **F:** 07-3036-5741

MySQL Support White Paper

Web Secure

PO Box 2726 Cairns, Qld 4870 Australia

P: 1300 73 83 01 **F:** 07-3036-5741

CONTENTS

Web Secure

PO Box 2726 Cairns, Qld 4870 Australia

P: 1300 73 83 01 **F:** 07-3036-5741

INTRODUCTION

This document summarises the key features of MySQL, the Relational Database Management System (RDMS). MySQL offers a fully-featured product that is fast, pragmatic and able to handle large databases efficiently.

What is MySQL?

MySQL is a Relational Database Management System. A relational database adds speed and flexibility, by storing data in separate tables rather than putting all the data in one area. These tables are linked by defined relations making it possible to combine data from several tables upon request. Using a RDMS means it is possible to add, access, and process the data stored in your database.

‘SQL’ stands for "Structured Query Language" - the most common standardised language used to access databases.

MySQL is Open Source software and is freely available at www.mysql.com. Open Source software means that the source code can easily be manipulated and modified by anyone. It is very simple to use.

Why use MySQL?

MySQL is very fast, reliable, and easy to use. MySQL also has a very practical set of features developed in close cooperation with its users. It is also Open Source and therefore freely accessible.

MySQL is used to access databases on the internet due to its connectivity, speed and security. It was originally developed to manage large databases at a much faster speed than the solutions that previously existed. MySQL has for several years, been thriving in the challenging areas of production.

Installing MySQL

Windows 2000 and NT 4.0 Installation

From -http://www.analysisandsolutions.com/code/mybasic.htm

If you don't already have MySQL on your system, you will need to download and install it.

1. Download MySQL-Win32 (http://www.mysql.com/downloads/index.html) version of the program.
2. Open up Windows Explorer.
3. Find the downloaded file and copy it into a temporary directory.
4. Double click on the file. At this point, if you have a zip utility, it should start. If you don't have a program to unzip files, you'll need to get one.
5. Once the files are extracted, go back into Explorer and double click on Setup.exe.
6. Follow the instructions within the installation program. Note: Do not change the installation directory from the default c:\mysql. If you want it installed somewhere else, the Win32 section of the MySQL manual (http://www.mysql.com/documentation/mysql/bychapter/manual_Installing.html#Win32) says to install it to c:\mysqlthen move it later. Once moved, you'll need to use the -basediroption when calling mysqldcommands.
7. Open up a MS-DOS Prompt window. The default location for starting such windows is: Start Menu | Programs | MS-DOS Prompt.
8. Switch to the c: drive if you're not already there: F:\>**c:**
9. Switch into the MySQL working directory: C:\>**cd mysql\bin**
10. Change the name of the daemon program: c:\mysql\bin**ren mysqldshareware.exe mysqld.exe**
11. Load the MySQL daemon: c:\mysql\bin**mysqld --install**
12. Close the DOS Prompt window: c:\mysql\bin**exit**
13. Open up the Services Manager
    a. NT 4.0: Start Menu | Settings | Control Panel | Services
    b. 2000: Start Menu | Programs | Administrative Tools | Services
14. Highlight the line with "MySql" on it.
15. If the "Startup" column says "Manual" or "Disabled, click on the "Startup" button, hit the "Automatic" radio button and click OK.
16. Click on the Start button.
17. Close the Services Manager and the Control Panel.

Note: Windows 95 or 98 do not have the ability to host a “service”, please do *not* run mysqld --install. You will need to run MySQL as a standalone application by executing the command mysqld --standalone.

Connecting to the Web Secure MySQL Server

When your database is initially created, you will be issued with a username and password to access your database. Web Secure gives you full control over your database however, it is MySQL that imposes some limits upon users and allows them to perform only certain functions).

To login through the MySQL client, you will need to use a connection string (which is available from the Technical Support Staff at Web Secure) similar to the following:

mysql <database> -h caelum.WebSecure.com.au -u <username> -p <password>

If you are using other MySQL enabled database management software to login, you will only require the hostname (caelum.WebSecure.com.au) and your username and password.

If you are unable to connect to your database, please contact support@Web Secure.com.au or call 1800 788 082.

Some simple MySQL queries Select Statement

select <item> from <table>;
select <item> from <table> where <variable> like <something>;

Insert Statement

insert into <table> values (variable1, variable2, variable3);

- where variable 1,2,3 are of the correct data type corresponding to the table elements. ie int = 1, char = 'Hello' etc."

Delete Statement

delete from <table> where <variable> = <data>;

Main Features of MySQL

Below are some of the important characteristics of MySQL and what it supports as listed at http://www.mysql.com/doc/F/e/Features.html:

- Fully multi-threaded using kernel threads. That means it can easily use multiple CPUs if available.
- Works on many different platforms. C, C++, Eiffel, Java, Perl, PHP, Python and Tcl APIs.

• Many column types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, and ENUMtypes.

• Very fast joins using an optimised one-sweep multi-join.

• Full operator and function support in the SELECT and WHEREparts of queries.

Example:

mysql> SELECT CONCAT(first_name, " ", last_name) FROM tbl_name WHERE income/dependents > 10000 AND age > 30;

• SQL functions are implemented through a highly optimised class library and are very fast. Usually there should not be any memory allocation at all after query initialisation.

• Supports SQL GROUP BY and ORDER BY clauses. Support for group functions (COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX() and MIN()).

• Support for LEFT OUTER JOIN and RIGHT OUTER JOINwith ANSI SQL and ODBC syntax.

• You can mix tables from different databases in the same query (as of Version 3.22).

• A privilege and password system that is very flexible and secure and allows host-based verification. Passwords are secure because all password traffic is encrypted when you connect to a server.

• ODBC (Open-DataBase-Connectivity) support for Win32 (with source). All ODBC 2.5 functions and many others. For example, you can use MS Access to connect to your **MySQL** server.

• Very fast B-tree disk tables with index compression.

• Up to 32 indexes per table are allowed. Each index may consist of 1 to 16 columns or parts of columns. The maximum index length is 500 bytes (this may be changed when compiling **MySQL**). An index may use a prefix of a CHARor VARCHARfield.

• Fixed-length and variable-length records.

• In-memory hash tables which are used as temporary tables.

• Handles large databases. We are using **MySQL** with some databases that contain 50,000,000 records and we know of users that uses MySQLwith 60,000 tables and about 5,000,000,000 rows

• All columns have default values. You can use INSERT to insert a subset of a table's columns; those columns that are not explicitly given values are set to their default values.

• Uses GNU Automake, Autoconf, and libtoolfor portability.

• Written in C and C++. Tested with a broad range of different compilers.

• A very fast thread-based memory allocation system.

• No memory leaks. Tested with a commercial memory leakage detector (purify).

• Includes myisamchk, a very fast utility for table checking, optimisation, and repair.

• Supports several different character sets, including ISO-8859-1 (Latin1) and mo re.

• All data are saved in the chosen character set. All comparisons for normal string columns are case insensitive.

• Sorting is done according to the chosen character set (the Swedish way by default). It is possible to change this when the **MySQL** server is started up. **MySQL** supports many different character sets that can be specified at compile and run time.

• Aliases on tables and columns are allowed as in the SQL92 standard.

• DELETE, INSERT, REPLACE, and UPDATE return how many rows were changed (affected). It is possible to return the number of rows matched instead by setting a flag when connecting to the server.

• Function names do not clash with table or column names. For example, ABSis a valid column name. The only restriction is that for a function call, no spaces are allowed between the function name and the `(' that follows it.

• All **MySQL** programs can be invoked with the --help or -? options to obtain online assistance.

• The server can provide error messages to clients in many languages.

• Clients may connect to the **MySQL** server using TCP/IP Sockets, Unix Sockets (Unixes), or Named Pipes (NT).

• The **MySQL**-specific SHOWcommand can be used to retrieve information about databases, tables, and indexes. The EXPLAIN command can be used to determine how the optimiser resolves a query.

MySQL Benchmarks

Most benchmarks (like the TCP ones) represent an SQL server's performance as a single number, often as transactions/second. These are almost worthless, since comparatively few users run applications that do the same thing as these benchmarks. There is usually no way to determine your application's performance from the numbers given by this type benchmark[1].

Licensing Agreement

Older versions of MySQL still require a stricter license. If you need a commercial MySQL license, because the GPL license does not suit your application, you can buy one. The following information is from:

http://www.mysql.com/support/arrangements/policy.html

For normal internal use, MySQL costs nothing.

A license is required if:

- You link a part of the MySQL that has a GPL Copyright to a program that is not free software (embedded usage of the MySQL server). In this case your application would also become GPL through the clause in the GPL license that acts as a virus. By licensing MySQL from us under a commercial license you will avoid this problem.

- You have a commercial application that only works with MySQL and ships the application with the MySQL server. This is viewed as linking even if it is done over the network.

- You have a distribution of MySQL and you do not provide the source code for your copy of the MySQL server, as defined in the GPL license.

A license is NOT required if:

- You do not need a license to include the client code in commercial programs. The client part of MySQL licensed with the LGPL GNU Library General Public License. The mysql command-line client includes code from the readline library that is under the GPL.

• If you use MySQL in a commercial, profitable context, it is requested that you further the development of MySQL by purchasing some level of support. It is reasonable to assume that if MySQL helps your business, in return then you should help MySQL.

If your use of MySQL does not require a license, but you like MySQL and want to encourage further development, you can purchase a license or MySQL support anyway.

For circumstances under which a MySQL license is required, you need a license per machine that runs the mysqld server. A multiple-CPU machine counts as a single machine, however, and there is no restriction on the number of MySQL servers that run on one machine, or on the number of clients concurrently connected to a server running on that machine.

MySQL Related Links

www.mysql.com- MySQL Reference Manual, OS Specific Drivers and Utilities
www.devshed.com/Server_Side/MySQL/ - Tutorials and sample code snippets
www.analysisandsolutions.com/code/mybasic.htm- MySQL Basics. A tutorial on installing MySQL on a Windows
machine.
www.linuxplanet.com/linuxplanet/tutorials/1046/1/ - Setting up a MySQL-based website (using perl).
http://hotwired.lycos.com/webmonkey/databases/tutorials/tutorial4.html - PHP/MySQL tutorial.

[1] From http://www2.linuxjournal.com/lj-issues/issue67/3609.html