



# MySQL vs. SQL Server

White Paper

---

## Abstract

This paper describes the differences between MySQL and Microsoft SQL Server 2000

## **Tometa creates custom software for you**

Tometa Software designs and develops robust software solutions for virtually all industries including in-house (vertical market) and retail software, some of which is on the shelves at your local software store. We focus our unique combination of creative, technical, and problem-solving skills on meeting our client's objectives. Because of our clarity of purpose, commitment to process, and broad professional skill sets, we are able to provide our clients with world-class solutions that are functionally superior and fully aligned with our client's strategic focus.

Balancing development speed, quality and cost is what we are all about. Tometa combines agile development practices with fixed pricing, so you know what the cost, end product, and delivery time table look like—up front. If we underestimate the effort, we complete the overrun on our dime. Simple as that. That's why large enterprise firms like Alcoa and NASDAQ choose Tometa.

Tometa's agile development expertise and low-overhead US location keep our prices comparable to offshore vendors – without offshore challenges. Using a fixed pricing model, we provide upfront visibility into a project's ultimate costs, end product and delivery schedule. Our clients like knowing that we have “skin in the game” – a fixed price that aligns our goals with yours, incenting us to get the job done right and fast.

Lastly, as a Microsoft Certified Gold Partner, Tometa Software, can customize its products or create custom web, client/server, and traditional applications. With programming experience in C#, C++, Visual Basic, CGI, HTML, RPG, Delphi, Java and many others; Tometa Software is uniquely positioned to meet your needs as a development firm.

[Check us out today](#)

When considering a utility for data management the two most popular choices are MySQL and SQL Server. Both are efficient at keeping your data organized and readily available through a user interface, but they differ drastically in many areas. In the following discussion we will review the major differences in detail to help clear the air about the most common question asked in data management discussions: “Which is better, MySQL or SQL Server?”

Some key questions to ask yourself in determining which data management tool is best suited for your needs are:

- How flexible do you need your data to be?
- How secure *must* your data be?
- Do you need support in utilizing your data management tools?

### **Standard Language Support**

An important aspect of database development is how to access your data using different standard based protocols. Both MySQL and SQL Server do a good job of supporting all major protocols for accessing their respective databases. A standard protocol is the “programming language” used for a program to communicate to a SQL database. The most common protocol is called tSQL, or Transact SQL. Transact SQL is a series of statements that a program can use to access data and create new tables in a SQL database. The statements can INSERT new records, DELETE old records and UPDATE existing records, along with a myriad of other functions.

The international standard for this programming language is referred to as ANSI SQL. ANSI stands for American National Standards Institute. ANSI is a collective governing body that determines what the standard language will be. For example, ANSI decides such things as the word INSERT will be used in the programming language as apposed to the work ADD to insert a new record into a database.

While both databases do a good job of supporting all major protocols, one of the most frequent complaints about MySQL is that it does not *completely* follow the ANSI SQL standard. This would not necessarily be any issue if your need for data storage never outgrows the capability of MySQL. However, if you data is constantly growing, there may be an eventual need to upgrade from MySQL to SQL Server to handle the vast amounts of data. In a situation like this, any application that has been written to manipulate a MySQL database will have to be revisited to be sure that the program is able to “communicate” with the new data store.

To give a better idea of the differences in MySQL and SQL Server, MySQL is geared more toward selecting data so is can be displayed, updated and saved

again. MySQL is weaker in the areas of inserting and deleting data. MySQL is an excellent choice, for other reasons that we will discuss later on, for data storage and referencing data. MySQL is not the best choice however, for creating, inserting and deleting data from tables on the fly.

For an Internet application, MySQL would be a good choice for tracking clients and creating dynamically populated pages with information from the database. For an application of moderate to large scale that is used for commerce of any kind, SQL Server is by far the better choice with more options and functionality to add, manipulate, update and delete data.

Following are some specific technical differences in MySQL and SQL Server when it comes down to the ANSI SQL standard:

- MySQL does not support Triggers, SQL Server does.
- MySQL does not support User Defined Functions, SQL Server does.
- MySQL does not have Cursor Support, SQL Server does.
- MySQL does not support Stored Procedures, SQL Server does.
- MySQL does not have FULL JOIN capabilities, SQL Server does.
- MySQL has limited Import/Export capabilities, SQL Server has full support for Import/Export.
- MySQL has limited Transaction Support while SQL Server offers full Transaction Support.

So, it becomes clear that SQL Server is an obvious choice for the company or individual that needs to have FULL control over their data to manipulate and change it as they need.

The following table shows a more complete comparison of the differences between MySQL and SQL Server:

Feature	MySQL 4.1	SQL Server 2000	Notes
SQL VIEW support	○	●	SQL VIEWS let administrators abstract database designs away from developers.
Triggers	○	●	Lack of triggers makes MySQL developers add extra logic to their front end and middle tier when the logic should go into the database.
Stored Procedures	○	●	Stored procedures are mechanisms for abstraction and security
User Defined Functions	○	●	User Defined Functions (UDFs) allow encapsulation of complex code into simple callable interfaces.
CURSOR Support	○	●	Lack of CURSOR support in MySQL increases network traffic and lowers app response time.

SQL SELECT	●	●	MySQL supports using a regular expression as a filter clause in a query.
XML Support	○	●	XML is a standards-based format for data. MySQL has no native support for XML.
FULL JOIN	○	●	MySQL does not support FULL JOIN
Referential Integrity	●	●	MySQL 4.0 supports referential integrity (RI) so long as the InnoDB table type is chosen.
Transaction Support	◐	●	MySQL's default installation does not support transactions. Transaction support requires InnoDB.
Full Text Support	●	●	
Import/Export Support	◐	●	
Replication Support	◐	●	
Auto Tuning	○	●	
Database Management Tools	◐	●	
Query Analysis Tools	◐	●	
Job Scheduling	○	●	
Profiling	○	●	
Online backup support	◐	●	
Clustering Support	◐	●	
Log Shipping	○	●	
Support for Storage Area Networks (SANs)	○	●	
Hot Backups / Incremental backups	◐	●	
Basic Security	◐	●	
Security Certifications	○	◐	
Security Tools	○	◐	
Security through stored procedures & views	○	●	
OLAP Services	○	●	
Data Mining	○	●	
Data Reporting	○	●	

## Security Support

Another major concern to business today is security. While data management can seem to be a mundane process at times, securing critical data from “the outside world” is an ever increasing and trying task.

While it is good to know that your database management system (MySQL or SQL Server) utilize security features it is very important to know that the application has been verified by a third party. SQL Server has been certified as C2 compliant, which ensures that the database has adequate security support for government applications.

Along with C2 certification, Microsoft Baseline Security Analyzer helps administrators ensure that their SQL Server installations are up to date with the most current patches and security features. MySQL has no equivalent tool to protect and ensure the same confidence in their platform.

While both MySQL and SQL Server support security measures within in their platforms, MySQL supports security via the SQL GRANT command. The MySQL GRANT command is limited to granting security at the table level. This means that if any portion of data in the table needs to be secure from any particular user, then the entire table has to be secured from that user. SQL Server supports security at the column level, meaning that any portion of data in a table can be secured from any particular user while allowing that same user to see other portions of the table data.

SQL Server also makes their database more secure by abstracting its data behind a layer of stored procedures. This ensures that developers never see how the actual data is represented. MySQL, as noted above, does not support stored procedures.

## Data Protection

Another security aspect is the protection from loss of data due to power failure. Database administrators in large environments employ the concept of “warm standby servers”. This is an additional server containing duplicate data that, by normal use, is not accessed until an event such as a power failure of the main server. Standby servers are kept in a ready state using a concept known as log shipping. Log shipping takes a regularly scheduled backup (sometimes at intervals of less than 10 seconds) of a database and stores that data on the duplicate standby server’s hardware.

MySQL lacks the capability to perform such a backup transaction in an online, or “live” environment. This feature is standard to SQL Server and includes tools to help the database administrator manage the details of the log shipping.

## Support and Service

When deciding on how you want to manage your data, you must also consider what kind of support you can expect to receive from the manufacturer of your data management application. After all, if you can't figure out how to make it work for you, you can't make any use of your data.

MySQL and SQL Server both offer support plans that are available from their respective vendors. Both applications offer some free support as well as some paid support options and plans.

MySQL claims to employ about 100 people worldwide, which makes up the entire company consisting of product development, support staff, sales staff, distribution, and clerical workers. With such a small staff dispersed over a wide area of expertises, this fact may worry some database administrators.

SQL Server, from Microsoft, has a significantly large support program and large staff backing the program.

## Pricing!

Now, let's move on to the biggest issue of them all. Pricing! Let's start with MySQL, how does FREE sound? Well, while MySQL is an open source database management tool, it does have its retractions on that \$0.00 price! MySQL can be used, for free, for any purpose whether its used to manage a large companies data or to spice up a hobbyist's web site.

The catch is, if your program, application or web site is using MySQL as the data management tool, the source for the program *must* be in the GPL (General Public License). This means that any program written to manage a MySQL database must be made public for all to see. This could pose some major conflicts with any business that is handling proprietary information of any kind. So, free is not always free!

SQL Server on the other hand, could take a small independent company's budget and eat it for lunch! However, the many outstanding features that are far and away above MySQL offset the price paid for SQL Server drastically.

## Conclusion

Now that we have taken a good look at the defining differences between MySQL and SQL Server, its plain to see that the final decision will come down to your needs. How flexible do you need your data to be? How secure *must* your data be? Do you need support in utilizing your data management tools? These are all questions that you will have to answer for yourself. And in answering them, you

will know which data management tool will be best suited for your personal, business or corporate needs.

From a database developer's standpoint, the choice is clear. SQL Server is the most sensible choice because of its rich features in manipulating, securing and managing data. Also, from a developer's stand point, MySQL's lack of support for the basic database features mean that development of an application to interface with the database will be both more costly and take longer to finalize. Too much code must be written in the user interface to manipulate the data first, before sending that data to the database. All this extra code costs time and money to develop and maintain.

---

This document is for informational purposes only. Tometa Software, Inc. MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2004 Tometa Software, Inc. All rights reserved.