



MySQL vs. PostgreSQL

White Paper

Abstract

This paper compares MySQL and PostgreSQL and identifies the differences in both of these popular database systems.

MySQL vs. PostgreSQL

Revised by Chris Rindal
QA Technician, Tometa Software, Inc.

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](#)

As business needs continue to expand and grow, so does the need for data storage that facilitates easy access and dissemination of information. We all hear about the advanced features and performance benefits of commercial database management systems - the likes of Oracle, SQL Server and DB2. But how about Open Source database engines?

MySQL and PostgreSQL are two of the most popular Open Source database management systems on offer today. They both boast an impressive array of features, data management tools and excellent end-user support.

This paper aims to give an overview of these features and compare the capabilities of each database engine. For a database application developer looking to take advantage or migrate to an open source platform, this is a good overview to choosing the right DBMS for your needs.

Topics in this Paper:

- **Features**
- **Support**
- **Ease of Use**
- **Stability**
- **Speed**
- **Licensing**

Features

A features comparison between MySQL and PostgreSQL is no easy feat. There is a fairly broad feature set with varying levels of support amongst the two management systems. The best way to identify the strengths and weaknesses is to compare the feature set we've come to expect from database systems.

Data Storage

PostgreSQL is fairly straightforward, using Postgre Storage System as the only storage mechanism. MySQL, on the other hand, offers a multitude of database storage systems, such as InnoDB and DBD. While this offers a greater choice of features, it can be a source of confusion and increased learning curve.

Database Integrity

Database integrity is a critical feature that differentiates between both databases. An important qualification that insures database integrity is the ACID test. ACID is an acronym that stands for atomicity, consistency, isolation and durability. These are four properties of a robust database system that are scoped to a transaction. A database that is ACID-compliant either commits the whole information to the database if the transaction is successful, or writes nothing in the event of an abnormal transaction.

Both Databases are ACID-compliant, but there is a marked difference in the scope of transaction functionality compliance. PostgreSQL is fully ACID-compliant and MySQL offers compliance only at the default table handler level. The standard table handler does not support consistency, durability or isolation. In order to insure database integrity you have to either download a version in which these handlers are specifically compiled or compile the database yourself after enabling these handlers.

When it comes to **Advanced Database Features**, PostgreSQL is by far the most sophisticated amongst the two. MySQL in its early days was built on the premise of the 20 / 80 rule: Only 20-percent of all SQL capabilities are required for 80 percent of database applications. As such, MySQL has sought to provide only a reasonable set of features, excluding those deemed surplus to requirements. PostgreSQL, on the other hand, offers a full set of advanced features and sophisticated methods.

For example, **Stored Procedures and Triggers** have only been supported by MySQL from version 5.0 and beyond. By contrast, it has been a standard support feature in PostgreSQL for some time. The query language, PL/PgSQL, is very much inline with commercial equivalents, such as Oracle's PL/SQL. There is also extensive support for other languages, such as Perl, Python and TCL in both safe and unsafe modes.

Primitive and large object **Data Types** are supported by both databases. Storing GIS (Geographic Information System) is also fully implemented. PostgreSQL has additional support for user-defined and network-aware data types.

Both database applications support **Replication**, in the form of single-master and multi-level replication scenarios. This is standard base-level functionality available with the distribution of the software. Additional replication methods for multi-slave and multi-master scenarios, as well as third-party replication are also available for PostgreSQL.

However, there are fundamental differences in the functions that support SQL statements. The basic statements, such as SELECT, UPDATE, INSERT and DELETE are the same, but once you step beyond the fundamentals, there is a different implementation, syntax and supported features. PostgreSQL also supports a richer SQL language than MySQL and this is no more evident than the support for SQL **Subqueries**. Subqueries can perform complex set manipulations and let data sets interact with each other in a sophisticated way.

Support

Being both open-source products, MySQL and PostgreSQL offer entrenched community support for end-users. These include mailing lists, active development forums and development articles. MySQL just edges it in the support department thanks to its popularity and wide-use amongst end-users.

For enterprise development, commercial distributions are available for both databases, offering a full set of support and training options.

On the MySQL front, commercial distribution is offered by MySQL AB. MySQL AB is the company guiding the development of MySQL and as such can make the software available under licenses other than GPL to suit end-user needs. Besides a set of fully-fledged support options, it also provides consulting, training and certification.

AbriaSoft offers a number of development and administration packages under its Merlin Server, which combines MySQL, PHP, Apache and Perl. The server runs under both Windows and Linux.

On the PostgreSQL front, there are a number of options you might consider. The most popular are PostgreSQL Inc. and NuSphere. PostgreSQL Inc. was set up to support the growth and market dominance of PostgreSQL as the SQL database of choice. The company provides financial help and code enhancements to the open source project. It offers a variety of support packages and maintains a training and certification program for developers and IT consultants.

NuSphere's UltraSQL offers a true native port of PostgreSQL under Windows, without any UNIX simulation. The server runs natively on Windows environments and offers a true, ANSI-SQL-compliant implementation of PostgreSQL. Alternatively, the product is bundled with PhpED, an integrated development environment, for a complete development solution.

Ease of Use

Ease of use is another thorny and potentially contentious issue. It all depends on your own needs and your development background.

If you're after a database to prop-up a Web blog with basic functionality, then MySQL is easier to use with just about the right functionality. PostgreSQL's extra features may be confusing and unnecessarily complex.

Conversely, for a complex database system PostgreSQL is easier to use. Hallmark features such as subqueries, unions, triggers and consistent procedures are essential for the development of such databases. Such functionality is complex and awkward to write in MySQL.

Stability

Both database systems claim a large level of stability in their new releases. To repeat yet another mantra, it all depends on your needs and the features you expect to use.

If you're running a website with a few hundred visitors a day, stability is a minor issue. Even Ms-Access can handle that sort of traffic and be stable!

Stability becomes an issue in heavy environments. In such a setting, both DBMS fare differently depending on the features in use. For example, MySQL would be more stable where replication is used since this feature has long been implemented in the database. Conversely, PostgreSQL's advanced set of features, such as stored procedures and subqueries, have been implemented for longer and as such are more likely to be stable.

Overall, with the increasing commoditisation of databases, stability can be taken for granted. It is no more an issue of stable software than the hardware it relies on.

Speed

MySQL has been designed to be a fast database. It is indeed faster than the fully-featured PostgreSQL. For example, MyISAM tables are very lightweight and make for an extremely speedy performance.

However, with both database systems converging from different directions – MySQL aiming to implement more advanced features, and PostgreSQL fine-tuning performance -, the gap is quickly closing as new releases are brought forward.

Licensing

Both MySQL and PostgreSQL are open source, but they are released under different license schemes. Understanding these licensing requirements and how they fulfil different needs is essential for incorporating the database systems into enterprise projects.

Let us first start with the MySQL, which operates a more complex two-tier licensing scheme. MySQL AB, the company that owns and develops MySQL, has two licenses:

General Public License (GPL): This license requires full public distribution of your application as well as the source code. You can also use the database under this license if you have no intention of ever distributing your application, either internally or externally.

Commercial License: The GPL license is a major stumbling block for commercial use as it proves a challenge especially when dealing with proprietary information. MySQL AB has made available this license for commercial users who don't want to make their code publicly available. The license also removes the requirement to distribute database drivers.

PostgreSQL is released under the much simpler Berkeley License (BSD) scheme. There is no requirement to distribute source code or drivers. The license allows the use of PostgreSQL or any of its derivatives as long as a copy of the BSD license is included.

Final Word: Consider the bigger picture

Both MySQL and PostgreSQL offer a good engine geared towards the lower to middle end of database systems. PostgreSQL has a full list of advanced features that have been stable for longer. As such, it is better suited for larger database systems. Over the long run, we expect MySQL to strongly stake a claim in the higher-end of the scale. The latest release has implemented a host of advanced features to support industry-standard database systems.

When it comes to making a decision about which database system to use, you cannot limit yourself to a simple set of features. There is a longer list of requirements that are equally crucial to the success of your database application. Consider documentation, support, pricing, stability and performance amongst other factors in light of your requirements. Only by taking all of these factors into account can you select the right database management system to meet your needs and the structure of your business.