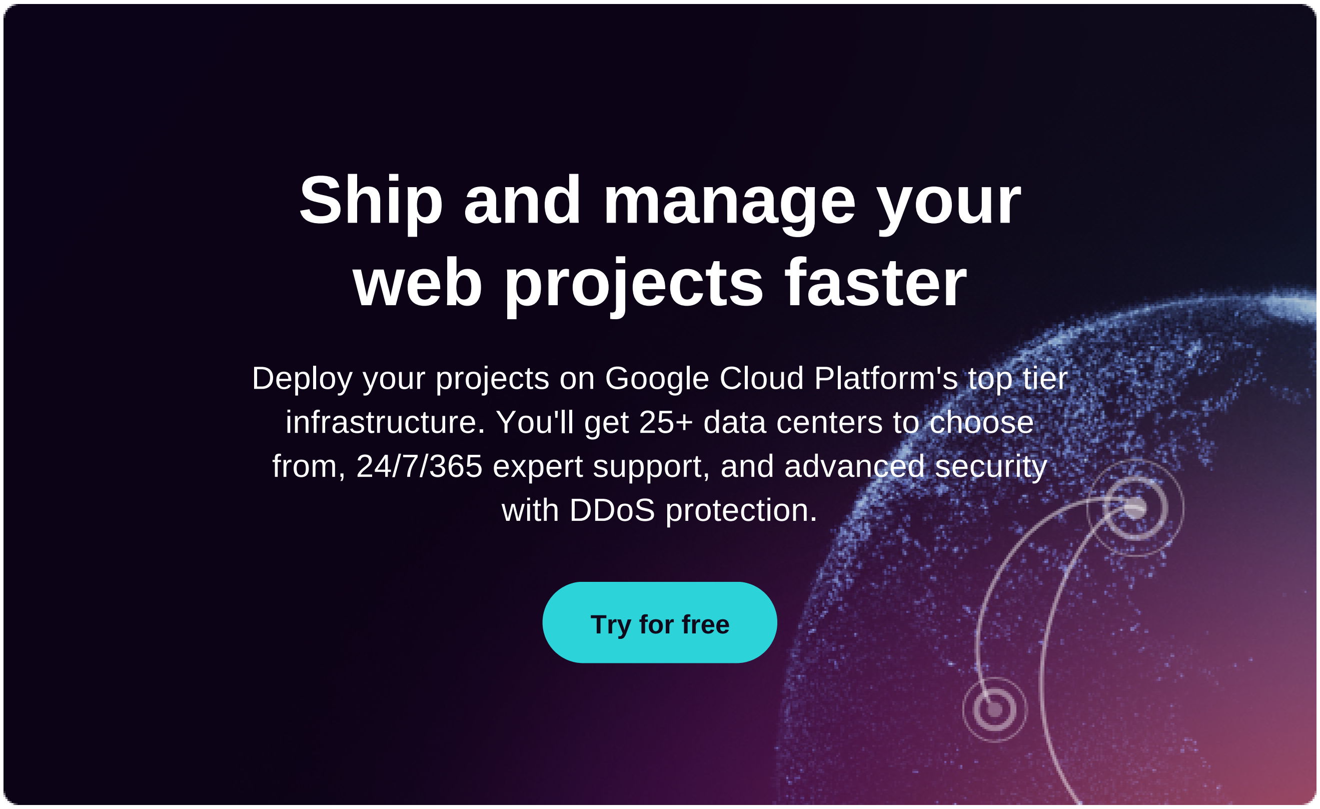
PostgreSQL vs MySQL: Explore Their 12 Critical Differences

Kinsta

Downloaded on: 18 June 2023



Data is essentially just a collection of different facts and observations. Over time, developers realized that managing data was not just an optional tracking system, but necessary as the world gradually became more connected via the internet.

Today, businesses leverage data to analyze potential clients, realize their potential, decrease risks, etc.

With the increase in data intake around the globe, there’s an ever-rising need for robust and flexible databases that can help manage data more efficiently. This article will examine the two most used open-source databases for WordPress and their differences: PostgreSQL vs MySQL.

But firstly, what are WordPress databases?

Let’s find out!

What Are WordPress Databases?

More than a few businesses use WordPress to host their websites, a whopping 43% of all the websites on the internet! In addition to that, it accounts for around 60% of all deployed Content Management Systems (CMS). Its easy-to-use layout makes it the perfect choice for beginners. While it isn’t necessary to have prior coding knowledge to use WordPress, it certainly does help to understand the different elements, including databases.

Without a WordPress database, your website won’t be able to function. The database system is essentially the backbone of your website. It ensures everything is tracked, from the content on your blog to the comments and changes made by different users. It even powers the website’s ability to load and execute.

An ideal database should be flexible, cost-friendly, and scalable. You can rest easy because there are a variety of open-source databases available which can help you track your data within WordPress. We’ll be focusing on PostgreSQL and MySQL for now.

What Is PostgreSQL?

— PostgreSQL logo (Image source: Uberconf)

PostgreSQL is an open-source, object-relational database management system. It is fully SQL-compliant and was built to be feature-rich. It’s also extendable, making it useful for anyone who needs enterprise tools. It was specifically designed for efficiency and can be integrated into almost any software.

PostgreSQL is object-oriented, making it possible to extend data types to create your custom types, and it has support for almost any database. This section will detail its history, features, and use cases.

History

40 years ago, a young pioneer, Michael Stonebraker, the leader of the Ingres project team, left Berkley to develop a proprietary version of Ingres. He then returned to Berkley and initiated a post-Ingres project which addressed several problems that the other databases encountered at the time.

That project, which we now know as PostgreSQL, was equipped with several features needed to support multiple “object-relational” data types, including support for rules to maintain a consistent relationship between the tables and the replication of data across servers. The first release of PostgreSQL formed version 6.0 on January 29, 1997. Since then, developers, support companies, and even volunteers have continued maintaining the database software under its free and open license.

Main Features

PostgreSQL has a lot to offer as a database management system. It has earned its reputation for feature-robustness, high reliability, performance, flexibility, and ease of replication.

Let’s look at what makes PostgreSQL an indispensable tool for your business.

Highly Reliable

PostgreSQL supports foreign keys, stored procedures, joins, and views in several languages. It includes various data types and supports the storage of large objects, including pictures, sounds, and videos. Since it is open-source, it’s backed by developers who provide an unmatched maintenance system by regularly trying to find bugs and improving the software.

It is also fault-tolerant due to the write-ahead logging feature, which makes it possible to support online backup and point-in-time recovery. We can support reverting to any time instant covered by the WAL data by installing a previous physical backup database.

Furthermore, the “physical backup” doesn’t have to be an immediate snapshot of the database state — if it was created in the past, replaying the WAL log for that specific time will resolve any internal inconsistencies.

Flexible

PostgreSQL is open-source, hence the code is freely available to modify cross-platform — it can be suited to work on any platform, including Windows, Solaris, OS X, and Linux. In addition to that, it can accommodate several users simultaneously, blocking only concurrent updates of the same row.

Extensibility

Extensibility is a software engineering principle that talks about future growth. PostgreSQL provides high extensibility as its operation is catalog-driven, i.e. information is stored in databases, columns, tables, etc. The just-in-time (JIT) compilation of expressions enables you to write your codes from different programming languages without recompiling your database and defining your data types. This ability to modify any operation spontaneously makes it uniquely suited to enforce new storage structures and applications rapidly.

Replication

PostgreSQL includes built-in synchronous replication, which ensures that the primary node would wait for each write until a duplicate node has written the data to its transaction log. The durability of the transaction can be specified per database, session, and user, irrespective of its synchronicity. This helps speed up the transaction because it doesn't need to confirm whether the transaction is reaching a synchronous standby, especially when some flows don't require these guarantees.

Use Cases

PostgreSQL is pretty much everywhere — it stands within the top five most used databases today, just behind MySQL. Major companies like Bloomberg, Goldman Sachs, and Nokia have PostgreSQL running at their backend.

PostgreSQL can be used by various industries and isn't limited to just one sector. Here are a couple of examples in which PostgreSQL can be used today.

- **Government GIS data**: PostgreSQL contains a powerful extension called “PostGIS.” This extension provides many functions that assist in processing different geometric forms like points, line strings and is optimized to reduce disk and memory footprint, thus improving query performance. Electricity, emergency services, and water infrastructure services are primarily dependent on GIS to locate crew members and direct them to accurate destinations, often under challenging conditions, thus, coming in handy for the government.
- **Manufacturing**: Many manufacturing industries demand a lot of data storage facilities at high levels of efficiency. PostgreSQL is a suitable choice for optimizing supply chain performance and storage. It is the preferred choice since it is ACID-compliant and can be configured for automatic failover, full redundancy, and almost-zero-downtime upgrades. Since Oracle’s new licensing policy has made it difficult for smaller businesses to sustain the cost of using Oracle, PostgreSQL is preferred.
- **Web technology**: PostgreSQL is not just a relational database; it can also serve as a NoSQL-style data store. You can have both — the relational and the document-oriented world — in a single product. It can function in many modern frameworks like Django (Python), Hibernate (Java), Ruby on Rails, PHP, etc. Due to its replication ability, websites can easily be scaled out to incorporate as many database servers as you need.
- **Scientific data**: Research and scientific projects can generate terabytes of data, which must be handled in the most practical way possible. PostgreSQL has excellent analytical capabilities and offers a powerful SQL engine, so processing large amounts of data won’t cause issues. PostgreSQL can also be extended easily. You can integrate Matlab and R to perform several mathematical and aggregation functions.

What Is MySQL?

— MySQL logo (Image source: Mecdata)

MySQL is a simple relational database system. It is very efficient and user-friendly, making it one of the most recognizable technologies. Using SQL, you can quickly grasp several Structured Query Language concepts (SQL) concepts to build powerful data storage systems. It's available for free and is open source, although it's also available under various proprietary licenses.

This section will discuss its history, its main features, and use cases. Let's dig in!

History

MySQL was founded by a Swedish company, MySQLAB, in 1995 by Michael “Monty” Widenius, Swedes David Axmark, and Allan Larsson. Sun Microsystems then acquired MySQLAB.

The purpose of MySQL was to provide efficient and reliable data management options to businesses and home users alike. Alpha and beta versions of the platform were released by 2000, and the majority were compatible with prominent platforms.

Around the same time, it went open-source. This allowed third-party developers to make significant changes to the system. However, going open-source meant a loss of revenue, but that was eventually recovered as MySQL started gaining popularity.

A whopping 2 million active installations were achieved by the end of 2001. To put it into perspective, that’s almost the population of Slovenia! At the beginning of 2002, the company expanded its business and opened its headquarters in the US. By then, the platform already had 3 million users, with revenue amounting to $6.5 million, and it’s only continued to gain in popularity since then.

Main Features

MySQL server is multithreaded, multitasking, and is designed to work on heavy-load production systems. It has transactional and non-transactional engines and is one of the easiest database systems to install. MySQL is well-liked among users because it’s easy to use, reliable, and fast.

Now that you’re aware of how MySQL came to be, let’s discuss some of its key features.

Ease of Use

MySQL rose to popularity because of its ease of use. It guarantees several features like triggers, stored procedures, and the like. It also includes various utilities like a backup program in case of crashes, mysqladmin, an administrative client, and a GUI (MySQL workbench) for management. For a beginner, it provides a wide range of options with a comprehensive GUI, helping to make it one of the top five databases used today.

High Flexibility

MySQL provides effective and secure transactions for large volume projects. It’s flexible enough to work in a dynamic environment. Since it’s open-source, the code is freely available and can be modified to your liking.

Reliability and Security

Just like PostgreSQL, MySQL also adheres to the ACID model. Hence, there is no need to worry while conducting transactions: It ensures data protection due to point-in-time recovery and auto-commit capabilities.

If the system crashes, it will revert to the last checkpoint, thus ensuring no data is lost. Plus, due to it being open-source, there’s a large community of developers that ensure that the system is working fine, extend their support in forums, and fix various bugs.

Additionally, it offers data integrity through support for foreign key constraints, averting data inconsistencies across tables. Since it has a password system, it delivers a secure interface and guarantees the password is verified based on the host before accessing the database. The password is encrypted while connected to the server.

High Performance

MySQL is pretty fast, reliable, and cheap because of its exceptional storage engine architecture. This means it can provide high performance without losing the vital functions of the software. It’s able to load quickly because of its cache memory.

Over time, MySQL has improved in its performance by ensuring features like B-tree disk tables with index compression, optimized nested-loop joins, and thread-based memory allocation. Row-level locking and constant reads in the storage engine offer additional performance benefits for multi-user concurrency.

Scalable

In addition to being free and open-source, MySQL programs can be written in many languages. The MySQL connector/NET allows developers to link their data to the database. The Connector/J interface delivers MySQL support for Java client programs that use JDBC associations. A client library written in C is available for clients written in C or C++ or any language that provides C bindings.

APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are also accessible. It is also one of the most preferred cross-platform database systems and can be used in Linux, Windows, Solarix, etc. All this shows that it is applicable in almost any software and operating system, which makes it highly scalable.

Open-Source License

MySQL is available for users under an open-source license. This enables users to freely use and modify the code to make it compatible with other domains.

Since it's open-source, it has a large amount of support from developers who make sure bugs and security issues are quickly fixed. MySQL has user groups, forums, and support to provide a built-in network to address issues as soon as possible while imparting education on the database.

Use Cases

MySQL proves to be useful for web applications as most servers rely on MySQL. Other than being used as a WordPress database, many non-WordPress businesses like Joomla, TYPO3, and Drupal also use MySQL as their primary database.

Here are a few use cases of MySQL that prove it to be a reliable and efficient database system:

- **OLTP transactions**: Transactions require speed and accuracy. MYSQL can be scaled to 1000s of queries per second with efficiency and ease. The transaction needs to ensure Atomicity, Consistency, Isolation, and Durability (ACID). MySQL also adheres to the ACID principles, making it safe for critical transactions. If a system fails during a transaction, it rolls back to a checkpoint.

- **LAMP open-source stack**: MySQL is essential to numerous applications operating on the LAMP open-source software stack (LAMP stands for Linux, Apache, MySQL, and PHP/Python/Perl). LAMP is a universal solution stack for web services and is widely regarded as the medium of choice for both dynamic websites and high-performance web applications.
- **E-commerce applications**: MySQL is one of the most prevalent transactional machines for eCommerce platforms. It’s beneficial for managing customer data, transactions, and product catalogs. In ecommerce solutions, MySQL is often used simultaneously with other, non-relational databases, including document and key-value stores for syncing order data, and storing non-product data.

PostgreSQL vs MySQL: Head-to-Head Comparison

If you’re unsure about the right database for your business, this section will help you choose the best path. While PostgreSQL and MySQL are handy, practical, and popular, it’s imperative to choose the database more tailored to your needs.

This section will deep-dive into the various distinctions between the two databases.

Syntax

When it comes to syntax, both Postgresql and MySQL are similar. Here’s what a select query would look like for both:

```
SELECT * FROM STUDENTS;
```

However, MySQL doesn’t support several subqueries, like “LIMIT” or “ALL.” It also doesn’t support standard SQL clauses like “INTERSECT” or “OUTER JOIN.”

MySQL isn’t as fully SQL-compliant as PostgreSQL, which *does* support all of the sub-queries mentioned above. If you need to use these subqueries frequently for your business, then PostgreSQL would be a more apt choice.

Languages Supported

PostgreSQL and MySQL support many of the same languages with a few differences.

PostgreSQL, on the other hand, offers support for a wider range of programming languages:

- C/ C++
- Delphi
- Erlang
- Go
- Java
- Javascript
- JSON (native since version 9.2)
- Lisp
- .NET
- Python
- R
- Tcl
- Other programming languages

Here’s a list of languages that MySQL supports:

- C/C++
- Delphi
- Erlang
- Go
- Java
- Lisp
- Node.js
- Perl
- PHP

- R

Speed

Speed is an integral factor when deciding on the best database for your business requirements. A fast database will not only ensure that your website runs faster, but it will also help alleviate the strain on your servers by pointing out unused data you can remove.

Both PostgreSQL and MySQL are famous for being some of the fastest DBMS solutions floating in the market. However, there’s no clear winner in this category. You can quite easily find benchmarks that recommend one database based on the configuration, the test, and the hardware. One might have the upper hand on concurrency, while the other might fare better on a single-core machine with little memory.

Ultimately, it comes down to how you use them. MySQL is generally known to be faster with read-only commands at the cost of concurrency, while PostgreSQL works better with read-write operations, massive datasets, and complicated queries.

Architecture

MySQL is a purely relational database, whereas PostgreSQL is an object-relational database. PostgreSQL offers more sophisticated data types, and lets objects inherit properties. On the flip side, it also makes it more complex to work with PostgreSQL. PostgreSQL houses a single, ACID-compliant storage engine. MySQL offers support for 15 different storage engines apart from its default storage engine, InnoDB. The vast array of storage engines allows you to quickly leverage them for other use cases.

PostgreSQL generates a new system process via memory allocation for every client connection established. This requires a lot of memory on systems with many client connections. On the other hand, MySQL utilizes a single process and maintains a single thread for every connection. This makes MySQL the more suitable choice for applications of less-than-enterprise scope.

Performance

PostgreSQL was built to be standards-compliant, feature-rich, and extendable. Previously, PostgreSQL performance was on an even keel — reads were usually slower than MySQL, but it could write large amounts of data more efficiently. On top of this, PostgreSQL handled concurrency better than MySQL.

The gap between their capabilities has significantly reduced in the last few years. MySQL is still pretty fast at reading data if you're using the old MyISAM engine. It has also been optimized to catch up to PostgreSQL regarding heavy data writes.

When selecting a suitable tool for your purposes, performance shouldn't be a binding factor for most garden-variety applications. Both PostgreSQL and MySQL are — mostly — equally performant.

Replication & Clustering

Replication is a process that lets developers replicate data from a database to its duplicate databases. This ensures that every user has the same level of information. Replication also brings various benefits like fault tolerance, scalability, automated backups, and the ability to perform long queries without affecting the primary cluster.

Info

In addition to database backups, some hosts offer full-site automated backups. Kinsta is no exception. Take a look at how our disaster recovery protocols keep you and your data safe.

Both MySQL and PostgreSQL support replication. PostgreSQL offers synchronous replication, which means that it has two databases running simultaneously, and the primary database is synced with the duplicate database. You can even perform synchronous and cascading replication with PostgreSQL. In MySQL, however, the replication is one-way asynchronous. This means that one database server acts as the primary one, and the others are replicas.

Both MySQL and PostgreSQL support clustering as well. Clustering leverages the shared storage to replicate an equal set of data to each node within an environment. This lets databases tolerate failures, owing to the redundancy created by duplicating data across various nodes in an environment.

Data & Table Structure

JSON support remains one of the leading NoSQL features incorporated by MySQL. In contrast, PostgreSQL supports user-defined types, arrays, hstore, and XML. The main benefit of having the ability to operate with more data types is increased functionality. For example, by accepting arrays as a data type, PostgreSQL can also provide host functions compatible with those arrays.

However, despite the advantages of using alternative formats to store data, it can be more complex to execute such data formats, given that they do not follow a long-standing benchmark. Therefore, components used in tandem with the database might not always adhere to PostgreSQL formats.

MySQL is only partly SQL-compliant in terms of SQL compliance because it doesn’t support all the features like the no check constraint. That said, it does provide a lot of extensions.

In contrast, PostgreSQL is more SQL-compliant than MySQL, supporting most of the primary SQL features — 160 out of 179 mandatory features, to be precise.

Extensibility

PostgreSQL is regarded as a highly extensible tool since it supports various advanced data types that one can’t find in MySQL. This would include network address types, native UUID, geometric/GIS, JSON which can be indexed, and timezone-aware timestamps. If this didn’t make PostgreSQL a clear winner for this round, you could add your operators, data types, and index types.

So, if your application is tackling unstructured data or any of the unique data types it has available, PostgreSQL might be the better suitor. However, if you only deal with basic numeric and character data types, both databases should work fine.

Indexes

To improve database performance, you can use indexes by speeding up SQL queries when tackling large data tables. Without indexes, queries would be slow and a major burden for the DBMS.

Both PostgreSQL and MySQL offer distinct indexing options. PostgreSQL index types include the following:

- Partial indexes that only arrange information from a section of the table
- B-tree indexes and hash indexes
- Expression indexes that generate an index resulting from express functions instead of column values

MySQL, on the other hand, offers the following index options:

- Indexes stored on R-trees, such as indexes found on spatial data types
- Indexes stored on B-trees, such as PRIMARY KEY, INDEX, FULLTEXT, and UNIQUE
- Inverted lists and hash indexes when utilizing FULLTEXT indexes

Security

Both PostgreSQL and MySQL support group and user management, and granting SQL privileges to various roles. MySQL supports native window services, PAM, and LDAP for user authentication, while PostgreSQL supports IP-based client authentication and filtering using Kerberos and PAM. So, the two databases are neck and neck in terms of security.

Support & Community

Both PostgreSQL and MySQL have helpful communities to provide support to users.

PostgreSQL boasts a large community of volunteers who offer free advice to users through mailing lists and IRC. On top of this, you can even purchase paid support through third-party providers. You can even troubleshoot by going through the various helpful PostgreSQL books and manuals on the market.

MySQL too has a large volunteer community that devotes its time to help you out with free recommendations and support. You can avail this kind of support on the Percona and MySQL websites. On top of the free community support, Oracle also offers 24/7 paid support with the commercial versions of all its products. Like PostgreSQL, you can also carry out your troubleshooting by delving into the numerous free and helpful MySQL guides, books, and tutorials.

To sum it up, support for PostgreSQL might be a little challenging because it requires more technical expertise to set up and use. Also, the number of PostgreSQL experts is lower than the number of MySQL experts at your disposal today. So, in terms of user support and ease of management, MySQL is slightly better.

PostgreSQL vs MySQL vs Alternatives

Of course, MySQL and PostgreSQL aren’t the only database choices you can work with or even your only two open-source database choices. Enough about PostgreSQL and MySQL. Let’s introduce some other alternatives that can give these two a run for their money!

1. MongoDB

— MongoDB logo (Image source: Kubirds)

MongoDB is a free-to-use, source-available, document-oriented, cross-platform database program. This NoSQL database program leverages JSON-like documents with optional schemes to function effectively. MongoDB ensures that you can ship and iterate 3-5x faster with their unified in-house query interface suitable for any use case and a flexible document data model.

MongoDB serves as the foundation for every industry, irrespective of whether you are building mission-critical applications or stretching the limits of customer experience. Here are a few key features of MongoDB that have helped establish it as a viable alternative to PostgreSQL and MySQL:

- **Sharding**: MongoDB allows its users to scale their applications horizontally via sharding, a method used to distribute large datasets across numerous data collections. MongoDB users can utilize a shard key (a primary key with single or various replicas) to ascertain the data distribution within a collection and partition the data into different ranges across shards.
- **Ad-hoc queries**: Ad-hoc queries are stand-in commands that offer different returns for implementing queries. MongoDB also supports regular expression (Regex), range query, and field searches.

- **File storage**: You can leverage MongoDB as a file system, called GridFS, which comes with load balancing and data replication features for multiple computers to store files. GridFS or grid file system consists of MongoDB drivers, which can be accessed with Lighttpd plugins and Nginx or the mongofiles utility.

2. MariaDB

— MariaDB logo (Image source: Docker Hub)

MariaDB is a commercially supported fork of the MySQL relational database management system whose purpose-built and pluggable storage engines support workloads that previously needed a wide variety of peculiar databases. You can deploy MariaDB in minutes for analytical, transactional, or hybrid use cases.

Boasting an illustrious clientele that consists of Nasdaq, Deutsche Bank, DBS Bank, ServiceNow, Verizon, and Walgreens (among others), MariaDB is known for delivering unparalleled operational agility without abandoning key enterprise features such as full SQL and ACID compliance.

Here are a few critical features of MariaDB that make it an indispensable tool:

- **Virtual columns**: The support for virtual columns is one of the critical features of MariaDB. Virtual columns can be used to execute calculations at the database level. When more than one app accesses one column, users don’t have to write the calculations in every app separately. Instead, the database does it on their behalf.
- **Database views**: Views are good database performance optimization features. MariaDB takes a different road than MySQL when involving virtual tables during the querying of a view.
- **Thread pooling**: Thread pooling helps accelerate MariaDB’s work when dealing with multiple database connections in your pipeline. Instead of opening a separate thread for each connection, thread pooling provides you with a pool of open threads.

PostgreSQL vs MySQL: Which Should You Choose?

To summarize the discussion, choosing between the two databases isn’t always straightforward. Since there are no wrong answers here, it boils down to context.

If you’re looking for a feature-rich database that can smoothly tackle voluminous databases and complex queries while allowing you to grow any application to enterprise scope, you should go with PostgreSQL.

On the other hand, if you’re a beginner looking for a database that’s easier to manage and set up while still being reliable, fast, and well understood, you might try MySQL.

If you simply can’t decide, one option is to take them both for test drives before making your final decision. You can download and use DevKinsta, our free local development tool, to try out MySQL, and a different local development tool or service to sample PostgreSQL.

Summary

In this article, we’ve discussed the primary differences between PostgreSQL vs MySQL. These included pivotal factors like speed, performance, syntax, extensibility, security, support and community, indexing, and architecture, among others, to help you make an educated decision regarding the tool that fits your unique business requirements.

We concluded that it’s a close fight between the two, with PostgreSQL and MySQL having distinct merits and challenges. The “right” choice will ultimately come down to you and how you plan to run your business.

Between PostgreSQL vs MySQL, which do you plan on using for your next project, and why? We’d love to hear your thoughts! Share them in the comments section below.