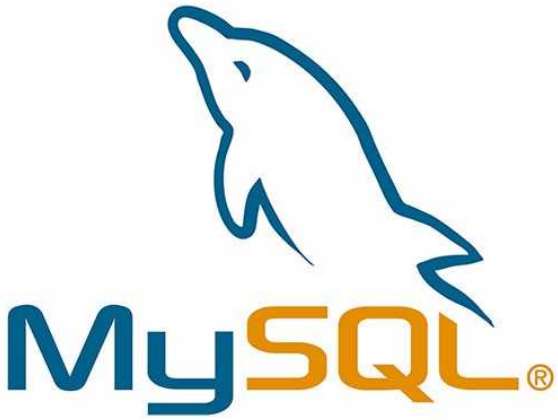

Features of MariaDB

MariaDB provides the same features of MySQL with some extensions. It is relatively new and advance.

Following is a list of features of MariaDB:

- MariaDB is licenced under GPL, LGPL, or BSD.
- MariaDB includes a wide selection of storage engines, including high-performance storage engines, for working with other RDBMS data sources.
- MariaDB uses a standard and popular querying language.
- MariaDB runs on a number of operating systems and supports a wide variety of programming languages.
- MariaDB offers support for PHP, one of the most popular web development languages.
- MariaDB offers Galera cluster technology.
- MariaDB also offers many operations and commands unavailable in MySQL, and eliminates/replaces features impacting performance negatively.

MariaDB Data Types

Following is a list of data types in MariaDB:

- String data types
- Numeric data types
- Date/time data types
- Large object data types

String Data Types

data type syntax	maximum size	explanation
char(size)	maximum size of 255 characters.	where size is the number of characters to store. fixed-length strings. space padded on right to equal size characters.
varchar(size)	maximum size of 255 characters.	where size is the number of characters to store. variable-length string.
tinytext(size)	maximum size of 255 characters.	where size is the number of characters to store.
text(size)	maximum size of 65,535 characters.	where size is the number of characters to store.
mediumtext(size)	maximum size of 16,777,215 characters.	where size is the number of characters to store.
longtext(size)	maximum size of 4gb or 4,294,967,295 characters.	where size is the number of characters to store.
binary(size)	maximum size of 255 characters.	where size is the number of binary characters to store. fixed-length strings. space padded on right to equal size characters.
varbinary(size)	maximum size of 255 characters.	where size is the number of characters to store. variable-length string.

Date/Time Data Types

data type syntax	maximum size	explanation
date	values range from '1000-01-01' to '9999-12-31'.	displayed as 'yyyy-mm-dd'.
datetime	values range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.	displayed as 'yyyy-mm-dd hh:mm:ss'.
timestamp(m)	values range from '1970-01-01 00:00:01' utc to '2038-01-19 03:14:07' utc.	displayed as 'yyyy-mm-dd hh:mm:ss'.
time	values range from '-838:59:59' to '838:59:59'.	displayed as 'hh:mm:ss'.
year[(2|4)]	year value as 2 digits or 4 digits.	default is 4 digits.

Numeric Data Types

data type syntax	maximum size
bit	very small integer value that is equivalent to tinyint(1). signed values range from -128 to 127. unsigned values range from 0 to 255.
tinyint(m)	very small integer value. signed values range from -128 to 127. unsigned values range from 0 to 255.
smallint(m)	small integer value. signed values range from -32768 to 32767. unsigned values range from 0 to 65535.
mediumint(m)	medium integer value. signed values range from -8388608 to 8388607. unsigned values range from 0 to 16777215.
int(m)	standard integer value. signed values range from -2147483648 to 2147483647. unsigned values range from 0 to 4294967295.
integer(m)	standard integer value. signed values range from -2147483648 to 2147483647. unsigned values range from 0 to 4294967295.
bigint(m)	big integer value. signed values range from -9223372036854775808 to 9223372036854775807. unsigned values range from 0 to 18446744073709551615.
decimal(m,d)	unpacked fixed point number. m defaults to 10, if not specified.d defaults to 0, if not specified.
dec(m,d)	unpacked fixed point number. m defaults to 10, if not specified. d defaults to 0, if not specified.
numeric(m,d)	unpacked fixed-point number. m defaults to 10, if not specified.d defaults to 0, if not specified.
fixed(m,d)	unpacked fixed-point number. m defaults to 10, if not specified. d defaults to 0, if not specified.
float(m,d)	single precision floating point number.
double(m,d)	double precision floating point number.
double precision(m,d)	double precision floating point number.
real(m,d)	double precision floating point number.
float(p)	floating point number.
bool	synonym for tinyint(1)
boolean	synonym for tinyint(1)

Large Object (LOB) Datatypes

data type syntax	maximum size
tinyblob	maximum size of 255 bytes.
blob(size)	maximum size of 65,535 bytes.
mediumblob	maximum size of 16,777,215 bytes.
longtext	maximum size of 4gb or 4,294,967,295 characters.

Who Uses These Databases?

MySQL: MySQL has generated a strong following since it was started in 1995. Some organizations that use MySQL include GitHub, US Navy, NASA, Tesla, Netflix, WeChat, Facebook, Zendesk, Twitter, Zappos, YouTube, Spotify, PayPal, Nokia, and more.

For MySQL, we can see names such as

You can check the full list here: https://www.mysql.com/customers/.

MariaDB: MariaDB is being used by many large corporations, Linux distributions, and more. Some organizations that use MariaDB include Google, Craigslist, Wikipedia, archlinux, Redhat, DBS, Suse, CentOS, Fedora, Ubuntu, 1&1, Ingenico and more.

Compatibility: What About Database Structure?

MySQL: MySQL is an open-source relational database management system (RDBMS). Just like all other relational databases, MySQL uses tables, constraints, triggers, roles, stored procedures and views as the core components that you work with. A table consists of rows, and each row contains a same set of columns. MySQL uses primary keys to uniquely identify each row (a.k.a record) in a table, and foreign keys to assure the referential integrity between two related tables.

MariaDB: Since MariaDB is a fork of MySQL, the database structure and indexes of MariaDB are the same as MySQL. This allows you to switch from MySQL to MariaDB without having to alter your applications since the data and data structures will not need to change.

This means that:

- data and table definition files are compatible
- client protocols, structures, and APIs are identical
- MySQL connectors will work with MariaDB without modification

Even the command line tools are similar to `mysqldump` and `mysqladmin` still having the original names, allowing MariaDB to be a drop-in replacement.

To make sure MariaDB maintains drop-in compatibility, the MariaDB developers do a monthly merge of the MariaDB code with the MySQL code. Even with this, there are some differences between MariaDB and MySQL that could cause some minor compatibility issues.

Bill Karwin, author of SQL Antipatterns: Avoiding the Pitfall, believes that MySQL still has a lot of potential and will eventually diverge from MariaDB. He says:

> As time goes on, MySQL develops more extensive features or changes to its internal architecture. They have more developers on staff than MariaDB, so they are making changes at a faster pace.

Gradually, MySQL have MariaDB diverged. A noteworthy example was the internal data dictionary under development for MySQL 8. This was a major change to the way metadata is stored and used within the server. MariaDB doesn't have an equivalent feature. This may mark the end of datafile-level compatibility between MySQL and MariaDB.

Comparing features – MySQL vs MariaDB

Many new and exciting features like Windows Functions, Roles or Common Table Expressions (CTE) are probably worth mentioning, but won't be mentioned in this article. We're all about comparing the two database engines, so, therefore, we'll only discuss features which are available only in one of them, to allow you, our readers, to determine the engine that works better for you.

Let's look into several features which are available only in one of the databases, exclusively:

1. **JSON datatype** – Starting version 5.7, MySQL supports a native JSON data type defined by RFC 7159 that enables efficient access to data in JSON (JavaScript Object Notation) documents.
   MariaDB decided not to implement this enhancement as they claim it's not part of the SQL standard. Instead, to support replication from MySQL, they only defined an alias for JSON, which is actually a LONGTEXT column. MariaDB claims there is no significant performance difference between the two, but no benchmarks were done recently to support that claim.
   It's worth noting that both MySQL and MariaDB offer different JSON related functions which allow easier access, parsing and retrieval of JSON data.
2. **Default authentication** – In MySQL 8.0, caching_sha2_password is the default authentication plugin rather than mysql_native_password. This enhancement should improve security by using the SHA-256 algorithm.
3. **MySQL Shell** – MySQL Shell is an advanced command-line client and code editor for MySQL. In addition to SQL, MySQL Shell also offers scripting capabilities for JavaScript and Python. You won't be able to access MariaDB servers using *mysqlsh*, as MariaDB doesn't support the MySQL X protocol.
4. **Encryption** – MySQL encrypts redo/undo logs (when configured to do so), while it doesn't encrypt temporary tablespace or binary logs. MariaDB, on the other hand, supports binary log encryption and temporary table encryption.
5. **Key Management –** MariaDB offers an AWS key management plugin out of the box. MySQL also provides several plugins for key management, but they're only available in the Enterprise edition.
6. **Sys schema** – MySQL 8.0 includes the sys schema, a set of objects that helps database administrators and software engineers interpret data collected by the Performance Schema. Sys schema objects can be used for optimization and diagnosis use cases. MariaDB doesn't have this enhancement included.
7. **Validate_password** – The validate_password plugin's goal is to test passwords and improve security. MySQL has this plugin enabled by default, while MariaDB doesn't.
8. **Super read-only** – MySQL enhances the read_only capabilities by providing the super read-only mode. If the read_only system variable is enabled, the server permits client updates only from users who have the SUPER privilege. If the super_read_only system variable is also enabled, the server prohibits client updates even from users who have SUPER. See the description of the read_only.

9. **Invisible Columns** – This feature, which is available on MariaDB, while not on MySQL, allows creating columns which aren't listed in the results of a SELECT * statement, nor do they need to be assigned a value in an INSERT statement when their name isn't mentioned in the statement.
10. **Threadpool** – MariaDB supports connection thread pools, which are most effective in situations where queries are relatively short and the load is CPU bound (OLTP workloads). On MySQL's community edition, the number of threads is static, which limits the flexibility in this situations. The enterprise plan of MySQL includes the threadpool capabilities.

Performance & Benchmarking

Over the years, many performance benchmark tests were executed on both MySQL and MariaDB engines. We don't believe there is one answer to the question "which is faster, MySQL or MariaDB?". It very much depends on the use case, the queries, the number of users and connections, and many other factors which should be considered.

These are the most recent benchmark tests we found online, which might provide some indication to which one performs better. Please make sure you read the details of each of the benchmark tests (how the benchmark was done, on which environment, which hardware, what was tested, what wasn't tested, etc.).

- MySQL 8.0 (InnoDB) and MariaDB 10.3.7 (MyRocks) benchmark test
- MariaDB 10.1 and MySQL 5.7 performance on commodity hardware
- MySQL 8.0 and MariaDB 10.3.5 performance and the UTF8 impact

Whichever major change you're planning: migrating from one database type (or engine) to another, from one OS to another, from on-premise servers to the cloud, I believe you should make sure you run your own tests, plan your own database benchmark process and perform the relevant stress testing. These are a few books we found on Amazon that may help you plan your next benchmark process:

- Database Benchmarking and Stress Testing: An Evidence-Based Approach to Decisions on Architecture and Technology – by Bert Scalzo
- High Performance MySQL: Optimization, Backups, and Replication – Chapter 2 – by Baron Schwartz, Peter Zaitsev, Vadim Tkachenko

Performance: Are Indexes Needed?

Indexes enhance database performance, as they allow the database server to find and retrieve specific rows much faster than without an index. But, indexes add a certain overhead to the database system as a whole, so they should be used sensibly.

Without an index, the database server must begin with the first row and then read through the entire table to find the relevant rows. The larger the table, the more costly operation.

MySQL and MariaDB: Most MySQL and MariaDB indexes (PRIMARY KEY, UNIQUE, INDEX, and FULLTEXT) are stored in B-trees. Exceptions include the indexes on spatial data types that use R-trees. MySQL also supports hash indexes and InnoDB engine uses inverted lists for FULLTEXT indexes.

Replication

Both databases provide the ability to replicate data from one server to another. The main difference we saw here is that most MariaDB versions will allow you to replicate to them, from MySQL databases, which means you can easily migrate MySQL databases to MariaDB. The other way around isn't that easy, as most MySQL versions won't allow replication from MariaDB servers.

Also, it's worth noting that MySQL GTID is different than MariaDB GTID, so once you replicate data from MySQL to MariaDB, the GTID data will be adjusted accordingly.

Few examples to the differences between the replication configurations:

- The default binlog format in MySQL is *row* based. In MariaDB, the default binlog format is *mixed*.
- Log_bin_compress – This feature determines whether or not the binary log can be compressed. This enhancement is unique to MariaDB and therefore isn't supported by MySQL.

Incompatibilities between MySQL and MariaDB

MariaDB's documentation lists hundreds of incompatibilities between MySQL and MariaDB databases, in different versions. The main conclusion from this documentation is that you can't rely on an easy migration from one database type to another.

Most database administrators hoped that MariaDB will be kept as a branch of MySQL, so it will be very easy to migrate between the two. For the last few versions, that's not the case anymore. For a long time now, MariaDB is actually a fork of MySQL, which means you need to put some thought when you migrate from one to another.

Syntax: How Are Their Queries Different?

MySQL queries are the same as MariaDB queries.

Selecting records from the customer table

MySQL:

```
SELECT * FROM customer;
```

MariaDB:

```
SELECT * FROM customer;
```

Inserting records into the customer table

MySQL:

```
INSERT INTO customer(cust_id, branch, status) VALUES ('appl01',
'main', 'A');
```

MariaDB:

```
INSERT INTO customer(cust_id, branch, status) VALUES ('appl01',
'main', 'A');
```

Updating records in the customer table

MySQL:

```
UPDATE customer SET branch="main" WHERE custage > 2;
```

MariaDB:

```
UPDATE customer SET branch="main" WHERE custage > 2;
```

Where (And How) Are These Databases Deployed?

MySQL: MySQL is written in C and C++ and has binaries for the following systems: Microsoft Windows, OS X, Linux, AIX, BSDi, FreeBSD, HP-UX, IRIX, NetBSD, Novell Netware, and many more.

To download MySQL go to the MySQL download page. There are installation instructions for Microsoft Windows, Linux, or OS X.

MariaDB: MariaDB is written in C, C++, Bash, and Perl and has binaries for the following systems: Microsoft Windows, Linux, OS X, FreeBSD, OpenBSD, Solaris, and many more.

Since MariaDB is designed to be a binary drop-in replacement for MySQL, you should be able to uninstall MySQL and then install MariaDB, and (assuming you're using the same version of the data files) be able to connect. Please note, you will need to run `mysql_upgrade` to complete the upgrade process.

To download MariaDB, go to the MariaDB downloads page. For Ubuntu, Red Hat, Fedora, CentOS,

or other Linux distributions, go to the download repository for your operating system. There are also installation instructions for Microsoft Windows, Linux, and OS X.

Storage engines

MariaDB supports more storage engines than MySQL. Said that, it’s not a matter of which database supports more storage engines, but rather which database supports the right storage engine for your requirements.

- Supported storage engines on MariaDB: XtraDB, InnoDB, MariaDB ColumnStore, Aria, Archive, Blackhole, Cassandra Storage Engine, Connect, CSV, FederatedX, Memory storage engine, Merge, Mroonga, MyISAM, MyRocks, QQGraph, Sequence Storage Engine, SphinxSE, Spider, TokuDB.
- Supported storage engines on MySQL – InnoDB, MyISAM, Memory, CSV, Archive, Blackhole, Merge, Federated, Example.

Deployed on Linux distributions by default

On some Linux distributions, when you install the MySQL database, you might end up actually installing the MariaDB database, as it’s the default in many Linux distributions (though not in all).

MariaDB will be installed by default on latest Red Hat Enterprise/CentOS/Fedora/Debian distributions. On the other hand, MySQL is still the default on other popular distributions such as Ubuntu.

Availability on cloud platforms

MariaDB is available as a service on Amazon Web Services (AWS), Microsoft Azure and Rackspace Cloud.

MySQL is available on all three platforms mentioned above, while also available on Google Cloud’s platform, as a managed service.

Therefore, if you are using GCP and would like your cloud provider to manage the service for you, you might have to consider using MySQL, unless you would like to install and manage MariaDB instances on your own.

Licensing

MariaDB Server is licensed as GPLv2, while MySQL has two licensing options – GPLv2 (for Community edition) and Enterprise.

The main difference between the two licenses for MySQL is the available features and support. While you receive the full-featured package when using MariaDB, that's not the case with MySQL. The community edition doesn't include features like the Threadpool, which can have a significant impact on the database and query performance.

Release-rate and updates

Usually, MariaDB has more frequent releases then MySQL. This reality has its pros and cons though. On the upside, features and bug fixes are released more frequently. On the other side, managing those MariaDB servers requires more updates to keep them up to do date at all times.

Technical Support

The MySQL support team, which includes both MySQL developers and support engineers, offer 24/7 support for customers. Oracle offers several support packages, including Extended support, Sustaining support and Premier support, depending on the customer's requirements. MariaDB's support team includes support engineers which are familiar and are experts with both MariaDB and MySQL databases (as many of the features were originally written by MySQL's team). They offer enterprise support for production systems, with 24/7 availability.

Ongoing Development

For MySQL, the exclusive developer is Oracle's MySQL team. On the other hand, MariaDB's development process is open for a public vote and mailing lists discussions. In addition, anyone can submit patches to MariaDB, which will be considered to be added to the main repository. Therefore, in a way, MariaDB is developed by the community, while MySQL is developed primarily by Oracle.

What Types Of Replication / Clustering Are Available?

Replication is a process that enables you to have multiple copies of the data copied automatically from 'master' to 'slave' databases. There are multiple benefits to this, and few of them being:

- backup
- spreading the load to improve performance
- analytics team can work on one of the slave databases, thus not hurting the performance of the main database in case of long-running and intensive queries.

Clustering, in the context of databases, refers to using shared storage and putting more database front-ends on it. The front end servers share an IP address and cluster network name that clients use to connect, and they decide between themselves who is currently in charge of serving clients requests.

MySQL: Replication in MySQL is one-way asynchronous replication where one server acts as a master and others as slaves. You can replicate all databases, selected databases or even selected tables within a database.

MySQL Cluster is a technology providing shared-nothing (no single point of failure) clustering and auto-sharding (partitioning) for the MySQL database management system.

Internally MySQL Cluster uses synchronous replication through a two-phase commit mechanism to guarantee that data is written to multiple nodes. This is in contrast to what is usually referred to as "MySQL Replication", which is asynchronous.

MariaDB: MariaDB offers master-master and master-slave replication as well. MariaDB uses the Galera Cluster for multi-master. As of MariaDB 10.1, Galera is included with MariaDB. Enabling clustering is as simple as activating the configuration parameters.

Who's Currently Behind The Databases?

In 2010, MySQL was acquired by the Oracle Corporation. At that time, one of the original developers, Michael “Monty” Widenius, felt that the Oracle Corporation had a conflict of interest between MySQL and their Oracle database.

In response to this, he decided to create a fork of the project named MariaDB. Since then, there's been a healthy competition between MySQL and MariaDB that has led to some great innovation. Since Oracle is backing MySQL and has a firm foundation, it continues to be the leader. However, MariaDB offers some compelling reasons for why people may want to switch databases.

MySQL: MySQL was originally started by MySQL AB in 1994 by a Swedish company that was created by David Axmark, Allan Larsson, and Michael “Monty” Widenius. The first version of MySQL was released in 1995. In 2008, Sun Microsystems purchased MySQL AB. In 2010, Sun Microsystems was acquired by Oracle.

MySQL is currently maintained by the Oracle Corporation.

MariaDB: On the day Oracle announced they had purchased MySQL, Michael “Monty” Widenius took several MySQL developers and started MariaDB, a fork of MySQL from that point.

Brian Wheeler from DevOps.com says:

> Widenius and many others in the open source community felt Oracle's ownership might be a conflict of interest since Oracle already had a competing closed source commercial database. They believed Oracle would be slow to further develop the MySQL database, given the potentially greater focus on the commercial Oracle database.

MariaDB enterprise is managed by the MariaDB Corporation AB. The community MariaDB server is managed by the MariaDB Foundation.

The MariaDB Foundation uses a community governance model. They chose to separate the open source and commercial sides of the business.

Brian Wheeler from DevOps.com says:

> Because of the separation between the open source and commercial sides of the foundation, its governance is seen by the open source community as positive. This is in contrast to Oracle, which has both its commercial Oracle database and the open source MySQL both under the same governance. Often this is perceived as a conflict of interest, especially when it comes to keeping MySQL up to date.

Who Provides Support?

MySQL: MySQL offers technical support services as part of Oracle's lifetime support. The support team contains MySQL developers and support engineers who offer 24/7 support as well as bug fixes, patches, and maintenance releases.

Oracle offers MySQL Premier Support, Extended Support, and Sustaining Support depending upon your needs.

MariaDB: MariaDB offers support engineers that are said to be experts in both MariaDB and MySQL. They offer 24/7 support with an enterprise subscription for mission-critical production systems.

Who Supplies Ongoing Development?

MySQL: Ongoing development is done by the Oracle Corporation, and they continue the development. Development decisions are not open to the public. Security releases come out every two months.

MariaDB: Where MySQL is developed by Oracle and decisions are not open to public discussion, MariaDB is developed with a different methodology. The development is open to the public where all development decisions can be debated and reviewed via a public mailing list. People can also submit patches for MariaDB. According to MariaDB, this methodology allows for more transparent and quicker security releases.

Who Maintains The Documentation?

MySQL: For MySQL, documentation is maintained by the Oracle Corporation.

MariaDB: For MariaDB, the main steward is the MariaDB Foundation, but other people can participate in development and documentation.

Is There An Active Community?

MySQL: MySQL is owned and managed by the Oracle Corporation. Oracle offers a Developer Zone on the MySQL website, which can be found at https://forums.mysql.com/. The site contains a variety of forums for running MySQL. You can view additional information at:

- MySQL Wiki
- Oracle MySQL Events
- MySQL Events
- List of MySQL user groups

MariaDB: Maria is developed by the open source community, allowing anyone to contribute. You can find additional information on how to connect with online community members, helping with documentation, development, events, and meetup groups at their Getting Involved page.

What About Database Connectors?

MySQL: MySQL offers a variety of database connectors including C, C++, Delphi, Perl, Java, Lua, .NET, Node.js, Python, PHP, Lisp, Go, R, D, and Erlang.

MariaDB: MariaDB offers a variety of database connectors including ADO.NET, C, C++, D, Java, JavaScript, ODBC, Perl, PHP, Python, Ruby, and Visual Studio plug-in.

Whether you choose MySQL, MariaDB, or both, Panoply has connectors for both databases. With a cloud based data warehouse architecture, it provides a no-coding solution without the need for data preparation or transformation. This allows you to consolidate all of your data from MySQL, MariaDB, cloud services, and applications into a single data management platform.

Which Database Is Right For Your Business?

MySQL: MySQL is a proven database that has a strong following around the globe. The Oracle Corporation continues to maintain, enhance, and support the product.

MariaDB: The MariaDB provides a drop-in replacement for organizations running MySQL. It is managed by the MariaDB Foundation, allowing people to contribute to the open source product and documentation.

MariaDB Oracle Compatibility

In addition, MariaDB Server 10.3 has Oracle compatibility (e.g., PL/SQL, sequences and data types), system-versioned tables for temporal queries (e.g., AS OF), user-defined aggregate functions, point-in-time rollback a la Oracle Flashback, distributed partitions via Spider, check constraints and set operators (INTERSECT/EXCEPT) to name a few.