Using MySQL and MariaDB

by Sophia Tutorial

WHAT'S COVERED

This tutorial explores the unique features of MySQL/MariaDB in three parts:

1. Introduction
2. MySQL History
3. Differences With ANSI SQL

1. Introduction

MySQL and MariaDB are two of the most widely used open-source databases. MySQL's popularity has much to do with WordPress, which runs more than a third of all websites around the world. WordPress is a content management system that uses MySQL as a database.

2. MySQL History

MySQL is an open-source project owned by Oracle. Oracle acquired MySQL when it acquired Sun Microsystems. The creators of MySQL worried that Oracle would simply kill MySQL because it was a direct competitor to their own enterprise solution, so they created MariaDB as a fork off of the code. The database structure and indexes of MariaDB are the same as MySQL, which allows organizations to switch between the two databases without any change in code.

MySQL is still being updated, and MariaDB is also constantly updated to ensure its compatibility with MySQL. This means that all aspects of the data, table definitions, APIs, protocols, file names, binaries, paths, ports, connectors, and more end up being the same.

As both databases are constantly being updated, it can be helpful to follow the differences and incompatibilities based on the version, which you can see here:

https://mariadb.com/kb/en/mariadb-vs-mysql-features/

3. Differences With ANSI SQL

There are some important differences to be aware of between ANSI SQL and MySQL/MariaDB. For example, if you access a column from a table to be updated, ANSI SQL uses the original value, whereas in

MySQL/MariaDB, the update uses the current value in the column in the order that they are set in. Consider the following statement:

UPDATE employee <br>
SET pay = pay + 1, new_pay = pay;<br>
Assume the original value for pay was 0. In ANSI SQL, the pay value will be set to 1, as it was set as pay = pay + 1. Since the original value for pay was 0, the pay is now 0 + 1, which is equal to 1. However, the value of new_pay would be set to 0, as that was the original value of pay.

In MySQL and MariaDB though, the value would be different. With the same statement, pay would again be set to 1, as it was set as pay = pay + 1. Since the original value for pay was 0, the pay is now 0 + 1 which is equal to 1. However, when new_pay is being set to pay, MySQL/MariaDB uses the updated value. The new_pay is set to 1 instead of 0, unlike with ANSI SQL. This may seem minor, but it could potentially create drastic issues.

Another key difference is the way that comments in the SQL are defined. In ANSI SQL, /* and */ are used for a block of comments. In MySQL and MariaDB, two dashes -- are used. This can be a bit confusing when you see this in a statement, especially when you're setting a value. For example, if we run the following in MySQL/MariaDB:

UPDATE employee <br>
SET pay = pay + 1 – 100;<br>
It would be the same as if you run just the following:

UPDATE employee <br>
SET pay = pay + 1;<br>
Anything that is after the -- is considered a comment and is thrown out.

SUMMARY

MySQL and MariaDB are two popular open-source databases that are very similar to one another.

Source: Authored by Vincent Tran