Using MySQL and MariaDB

by Sophia

WHAT'S COVERED

This lesson explores the unique features of MySQL/MariaDB, in three parts. Specifically, this lesson will cover:

1. Introduction

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

2. MySQL and MariaDB History

MySQL is a relational database management system in use worldwide, with a rich development and evolution history. It was originally developed by Michael Widenius and David Axmark for MySQL AB in 1994. The "My" in MySQL is a reference to co-founder Michael Widenius's older daughter, My, and the "Maria" in MariaDB is his younger daughter's name.

IN CONTEXT

MySQL was first released in 1995. Due to its ease of use, high performance, and low cost, it quickly became popular. MySQL was originally released under a dual-license model, offering both open-source and commercial licenses. In particular, LAMP (Linux, Apache, MySQL, PHP) stacks for running websites became popular due to MySQL's rapid popularity.

Sun Microsystems' acquisition of MySQL AB in 2008 led to Oracle Corporation's acquisition of Sun

Microsystems in 2010, making MySQL part of Oracle's portfolio. The open-source project MySQL has so far been actively developed and maintained under Oracle's ownership, despite some concerns about its future. However, Oracle does offer a commercial version of MySQL called MySQL Enterprise Edition that includes professional support services.

As a result of some concerns, MySQL was forked to MariaDB. It was feared that Oracle's ownership would lead to changes to MySQL's open-source nature, development direction, and licensing terms. To address these concerns, Michael “Monty” Widenius, one of the original developers of MySQL, created MariaDB as a drop-in replacement. The MariaDB platform was designed to remain true to the open-source philosophy, providing users with a more transparent, community-driven, and inclusive development process. Forking MySQL, MariaDB aimed to conserve the principles of collaborative development and community-driven innovation while continuing the legacy of an open, free, and accessible database system.

MariaDB also sought to improve MySQL's capabilities. The new features, optimizations, and performance improvements made it often superior to MySQL. MariaDB releases updates more frequently and fixes bugs and security issues more promptly, addressing user needs. Using an agile approach, users can benefit from the newest enhancements as soon as they are released. Today's MariaDB is a community-centric, open-source database solution that offers powerful, reliable, and viable alternatives to MySQL. Although MariaDB and MySQL split, both remain closely related, providing compatibility and familiarity for users switching between the two. Whereas MySQL is owned by Oracle, MariaDB has no official owner. The MariaDB Foundation develops and maintains the MariaDB database management system but does not own it. The MariaDB Foundation is an independent organization that was established to ensure the open-source nature of the MariaDB project and to provide a governance structure for its development.

Millions of developers and organizations around the world today use MySQL and MariaDB. As a result of their popularity, they have become an integral part of many web applications, content management systems, e-commerce platforms, and many other applications. Due to their performance, scalability, ease of use, and active community support, MySQL and MariaDB have remained relevant and widely adopted over the years.

3. Comparison to ANSI SQL

The ANSI SQL standard defines a common syntax and semantics for querying, modifying, and managing relational databases, ensuring portability and consistency across different database management systems. While MySQL implements some ANSI SQL standards, extensions, and proprietary features, it is a specific relational database management system.

The ANSI SQL standard defines a set of rules for how SQL queries and operations should be structured. Even though MySQL strives to be ANSI SQL-compliant, its feature set is not identical. It contains extra features not present in the standard, and it offers slight variations of certain SQL commands.

A DBMS offering MySQL's features and extensions beyond ANSI SQL is standard practice. Every database has its own set of features and extensions that conflict with other databases. The extensions are unique to MySQL and may or may not be present in other database systems following ANSI SQL.

KEY CONCEPT

There may be differences in how specific functions or operations are implemented in MySQL compared to other ANSI SQL-compliant databases. Performance and query optimization can be affected by these differences.

ANSI SQL defines standard data types, but individual database systems may support additional data types or handle certain data types differently. Some data types are unique to MySQL, and some are not standard. Different database systems, including MySQL, implement transactional behavior and concurrency control mechanisms differently.

Despite these differences, developers can write SQL queries in MySQL that are ANSI SQL-compliant, which makes the code more portable and compatible with other databases. However, if you use MySQL-specific extensions or features, the code may not be easily transferable to other database platforms. The MySQL standard differs from ANSI SQL, so developers need to know these differences. The goal is to write SQL code that is efficient, compatible, and maintainable.

Some of the key differences between ANSI and MySQL/MariaDB include:

- Data types: MySQL/MariaDB supports all the same data types as ANSI SQL, but there are a few minor naming differences. For example, ANSI SQL uses INTEGER, and MySQL/MariaDB uses INT.
- Storage engines: MySQL/MariaDB supports multiple storage engines, each with its own features. The ability to choose a storage engine is a MySQL-specific feature.
- MySQL Workbench: MySQL provides a graphical user interface called MySQL Workbench for database design, development, and administration. It can also be used with MariaDB.
- User-defined values: MySQL/MariaDB allows the use of user-defined variables, which are specific to MySQL.
- Full-text search: MySQL/MariaDB has robust full-text search capabilities that go beyond ANSI SQL.
- LIMIT and OFFSET: MySQL/MariaDB allows the use of LIMIT and OFFSET clauses with the UPDATE and DELETE statements; ANSI SQL does not.
- SHOW and DESCRIBE: MySQL/MariaDB has specific extensions to the SHOW and DESCRIBE statements, providing additional information about databases, tables, and indexes.
- Non-standard date and time functions: MySQL/MariaDB includes some date and time functions that are not part of the ANSI SQL standard.

IN CONTEXT

There are other small differences not covered in this lesson. For example, here's a seemingly minor difference that can potentially create drastic issues. 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
SET pay = pay + 1, new_pay = pay;
```

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.

SUMMARY

In this lesson, in the **introduction**, you learned that both MySQL and MariaDB are open-source relational database management systems (RDBMS). You explored the **history of MySQL and MariaDB**, learning that the MySQL database was originally developed in 1994 by Michael Widenius and David Axmark at MySQL AB. Several qualities contributed to its popularity, including ease of use, performance, and compatibility with different platforms. Eventually, Sun Microsystems and Oracle Corporation acquired MySQL, raising concerns about the open-source project's future. In response to this, Michael "Monty" Widenius, one of MySQL's original developers, created MariaDB, a fork of MySQL. The goal of MariaDB was to serve as a drop-in replacement for MySQL while maintaining an open-source philosophy and community development. A number of additional features, improved performance, and open-source commitments are included in MariaDB. With its community-centric focus, it quickly gained traction and became considered an attractive alternative.

You learned that the syntax of MySQL and MariaDB is similar, so applications designed for one database will work seamlessly with the other. You also learned about the **differences between MySQL/MariaDB and ANSI SQL**. Web applications, content management systems, and other projects widely use both RDBMS. There are many advantages to using MySQL, including its robustness, popularity, and Oracle Corporation's active support. MariaDB has seen rapid adoption, particularly among developers who appreciate its community-driven approach, frequent updates, and transparency. With competitive features, performance, and support for many use cases, MySQL and MariaDB are evolving. There are advantages and disadvantages to both, depending on individual preferences, needs, and whether one favors the backing of a large corporation (MySQL) or a vibrant open-source community (MariaDB).

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND Faithe Wempen (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.