**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN**

---------



**BÁO CÁO ĐỒ ÁN**

**TÌM HIỂU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MARIADB**

**Sinh viên thực hiện:**

Trần Trọng Nhân - 21522411

Nguyễn Thị Nhàn - 21522405

**Giảng viên:**

TS. Nguyễn Gia Tuấn Anh

GV. Trần Quốc Khánh

Thành phố Hồ Chí Minh, tháng 6 năm 2023

**BÁO CÁO TÓM TẮT**

**1. Tiêu đề báo cáo: TÌM HIỂU HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MARIADB**

**2.** **Danh sách thành viên**

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Ghi chú** |
| 21522411 | Trần Trọng Nhân |  |
| 21522405 | Nguyễn Thị Nhàn |  |

**3.** **Nội dung chi tiết**

***Nội dung 1: Giới thiệu***

+ Đề mục 1: Một vài nét về MariaDB

+ Đề mục 2: Nhà sáng lập và phát triển

+ Đề mục 3: Các phiên bản của MariaDB

***Nội dung 2: Các tính năng cơ bản***

+ Đề mục 1: SQL

+ Đề mục 2: NoSQL

***Nội dung 3: Các tính năng nâng cao***

+ Đề mục 1: Phân quyền

+ Đề mục 2: Xác thực

+ Đề mục 3: Backup và restore

+ Đề mục 4: Import và export

***Nội dung 4: So sánh MariaDB với MySQL***

***Nội dung 5: Kết luận và hướng phát triển***

+ Đề mục 1: Kết luận

+ Đề mục 2: Hướng phát triển

**4. Phân công công việc**

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Nội dung được phân công** |
| 21522411 | Trần Trọng Nhân | Nội dung 2: mục 2.2  Nội dung 3: mục 3.1, mục 3.2  Nội dung 4  Nội dung 5 |
| 21522405 | Nguyễn Thị Nhàn | Nội dung 1 Nội dung 2: mục 2.1  Nội dung 3: mục 3.3, mục 3.4  Nội dung 4 |

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 4](#_Toc137994189)

[1. Một vài nét về MariaDB 4](#_Toc137994190)

[2. Nhà sáng lập và phát triển 4](#_Toc137994191)

[3. Các phiên bản của MariaDB 6](#_Toc137994192)

[CHƯƠNG 2: CÁC TÍNH NĂNG CƠ BẢN 7](#_Toc137994193)

[1. SQL 7](#_Toc137994194)

[2. NoSQL 10](#_Toc137994195)

[CHƯƠNG 3: CÁC TÍNH NĂNG NÂNG CAO 13](#_Toc137994196)

[1. Phân quyền 13](#_Toc137994197)

[2. Xác thực 14](#_Toc137994198)

[3. Backup và restore 15](#_Toc137994199)

[4. Import và export 15](#_Toc137994200)

[CHƯƠNG 4: SO SÁNH MARIADB VỚI MYSQL 17](#_Toc137994201)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 18](#_Toc137994205)

[1. Kết luận 18](#_Toc137994206)

[2. Hướng phát triển 18](#_Toc137994207)

**DANH MỤC HÌNH VẼ**

[Hình 1. Logo chính thức của MariaDB 4](#_Toc137995784)

[Hình 2. Michael “Monty” Widenius, tác giả của MariaDB 5](#_Toc137995785)

[Hình 3. MariaDB Foundation, chủ sở hữu của MariaDB 5](#_Toc137995786)

[Hình 4. So sánh giữa hai phiên bản MariaDB Enterprise và SkySQL 6](#_Toc137995787)

# **CHƯƠNG 1: GIỚI THIỆU**

## **Một vài nét về MariaDB**

MariaDB là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, là một rẽ nhánh của MySQL được hình thành vào năm 2009. MariaDB là phiên bản cải tiến tương thích ngược của MySQL, đi kèm với nhiều tính năng có khả năng sẵn có khác nhau và nhiều cải tiến về bảo mật và thực thi mà MySQL chưa thể đáp ứng. MariaDB hỗ trợ các tính năng giống như MySQL, đồng thời cung cấp các tính năng bổ sung.

* MariaDB được viết bằng C, C++ và hỗ trợ nhiều ngôn ngữ lập trình, bao gồm C, C#, Java, Python, PHP và Perl. MariaDB cũng hỗ trợ tất cả các hệ điều hành chính, bao gồm Windows, Linux và macOS.
* MariaDB là sự kết hợp của cả SQL và NoSQL, hỗ trợ các chức năng liên quan đến NoSQL như CONNECT Storage Engine, Dynamic Columns, JSON Functions...
* Mô hình quan hệ được sử dụng chủ yếu trong hệ quản trị cơ sở dữ liệu MariaDB, ngoài ra còn hỗ trợ mô hình mạng và mô hình phân cấp.



Hình 1. Logo chính thức của MariaDB

## **Nhà sáng lập và phát triển**

MariaDB ban đầu là một dự án phụ hoặc phát triển thêm của dự án MySQL được dẫn dắt bởi Michael “Monty” Widenius. Ông cũng là một trong những nhà phát triển chính của MySQL vào năm 1994 và đồng sáng lập công ty Thụy Điển MySQL AB vào năm 1995. Công ty này đã được SUN Microsystems tiếp quản vào năm 2008 và sau đó là Oracle vào năm 2010. Sau khi công ty bị thu mua, ông đã rời đi và thành lập Monty Program Ab (nay là MariaDB Corporation), bắt đầu phát triển MariaDB, tên gọi MariaDB được đặt tên theo tên con gái út của Widenius – Maria. Hiện MariaDB được quản lí bởi MariaDB Foundation.



Hình 2. Michael “Monty” Widenius, tác giả của MariaDB



Hình 3. MariaDB Foundation, chủ sở hữu của MariaDB

## **Các phiên bản của MariaDB**

MariaDB Community Server được phát hành theo Giấy phép Công cộng GNU v2 và được đảm bảo là nguồn mở và miễn phí. MariaDB cũng hỗ trợ các phiên bản Enterprise và Cloud.

* MariaDB Enterprise bao gồm MariaDB MaxScale, trình kết nối ứng dụng và tích hợp, công cụ quản lý và hỗ trợ kỹ thuật. Có thể đăng ký MariaDB Enterprise bằng cách yêu cầu báo giá tùy chỉnh.
* MariaDB SkySQL là phiên bản Cloud của MariaDB, cung cấp tính sẵn sàng cao, khả năng mở rộng và bảo mật đi kèm với môi trường đám mây. SkySQL có tính năng hỗ trợ cho nhiều khối lượng công việc, khắc phục thảm họa và giám sát chủ động. MariaDB SkySQL đi kèm với chi phí $500 và bắt đầu tính từ $0,1318 mỗi giờ.



Hình 4. So sánh giữa hai phiên bản MariaDB Enterprise và SkySQL

# **CHƯƠNG 2: CÁC TÍNH NĂNG CƠ BẢN**

## **SQL**

MariaDB có các câu lệnh SQL gần giống như các cơ sở dữ liệu quan hệ khác. Sau đây là một số câu lệnh cơ bản:

* CREATE DATABASE: Tạo cơ sở dữ liệu mới.

*Cú pháp:*

**CREATE DATABASE [IF NOT EXISTS] database\_name;**

* USE: Sử dụng cơ sở dữ liệu hiện có.

*Cú pháp:*

**USE database\_name;**

* CREATE TABLE: Tạo bảng dữ liệu mới.

*Cú pháp:*

**CREATE TABLE [IF NOT EXISTS] table\_name(**

**column\_1\_definition,**

**column\_2\_definition,**

**...,**

**table\_constraints**

**) ENGINE=storage\_engine;**

* INSERT: Thêm dữ liệu vào bảng.

*Cú pháp:*

**INSERT INTO**

**table\_name(column\_list)**

**VALUES**

**(value\_list\_1),**

**(value\_list\_2),**

**(value\_list\_3),**

**...;**

* DROP: Xoá.

1. Xóa bảng dữ liệu.

*Cú pháp:*

**DROP TABLE [IF EXISTS] table\_name;**

1. Xóa cơ sở dữ liệu.

*Cú pháp:*

**DROP DATABASE [IF EXISTS] database\_name;**

* ALTER TABLE: Sửa bảng dữ liệu.

1. Thêm cột mới vào bảng.

*Cú pháp:*

**ALTER TABLE table\_name**

**ADD [COLUMN] new\_column\_name column\_definition**

**[FIRST | AFTER column\_name],**

**ADD [COLUMN] new\_column\_name column\_definition**

**[FIRST | AFTER column\_name],**

**...;**

1. Sửa cột trong bảng.

*Cú pháp:*

**ALTER TABLE table\_name**

**MODIFY [COLUMN] column\_name column\_definition**

**[FIRST | AFTER column\_name],**

**MODIFY [COLUMN] column\_name column\_definition**

**[FIRST | AFTER column\_name],**

**...;**

1. Xóa cột khỏi bảng.

*Cú pháp:*

**ALTER TABLE table\_name DROP [COLUMN] column\_name;**

1. Đổi tên cột trong bảng.

*Cú pháp:*

**ALTER TABLE table\_name**

**CHANGE [COLUMN] original\_name new\_name column\_definition**

**[FIRST | AFTER column\_name];**

1. Đổi tên bảng.

*Cú pháp:*

**ALTER TABLE table\_name**

**RENAME [TO] new\_table\_name;**

1. Thêm khóa chính.

*Cú pháp:*

**ALTER TABLE table\_name**

**ADD CONSTRAINT [pk\_constraint\_name]**

**PRIMARY KEY (column\_list);**

1. Thêm khóa ngoại.

*Cú pháp:*

**ALTER TABLE table\_name**

**ADD CONSTRAINT [fk\_constraint\_name]**

**FOREIGN KEY [fk\_name](column\_list) REFERENCES parent\_table(column\_list);**

* UPDATE: Cập nhật bảng dữ liệu.

*Cú pháp:*

**UPDATE table\_name**

**SET column1 = value1,**

**column2 = value2,**

**...**

**[WHERE search\_condition];**

* DELETE: Xóa dòng dữ liệu.

*Cú pháp:*

**DELETE FROM table\_name**

**[WHERE search\_condition];**

* SELECT: Truy vấn dữ liệu.

*Cú pháp:*

**SELECT** **column1,**

**column2,**

**…**

**FROM table\_name**

**[WHERE** **search\_condition]**

**[GROUP BY** **column\_name]** **[ASC | DESC]**

**[HAVING search\_condition]**

**[ORDER BY column\_name] [ASC | DESC];**

* **Kịch bản demo:** Sử dụng các câu lệnh SQL cơ bản làm việc trên cơ sở dữ liệu có sẵn.
* **Link video demo:** [**https://drive.google.com/file/d/1hQ16AToOEi1524rDAIMPioKRiLWEte-p/view?usp=drive\_link**](https://drive.google.com/file/d/1hQ16AToOEi1524rDAIMPioKRiLWEte-p/view?usp=drive_link)

## **NoSQL**

MariaDB có cung cấp các tính năng liên quan đến NoSQL. Sau đây là một số câu lệnh cơ bản về các chức năng JSON:

* JSON\_INSERT: Thêm dữ liệu vào tài liệu JSON, trả về tài liệu JSON mới sau khi thêm, hoặc trả về NULL nếu bất kỳ đối số nào là NULL.

*Cú pháp:*

**JSON\_INSERT(json\_doc, path, val[, path, val] ...)**

* JSON\_ARRAY\_INSERT: Thêm giá trị mới vào vị trí xác định trong mảng được chỉ định trong tài liệu JSON, trả về tài liệu JSON mới sau khi thêm, hoặc trả về NULL nếu bất kỳ đối số nào là NULL.

*Cú pháp:*

**JSON\_ARRAY\_INSERT(json\_doc, path, value[, path, value] ...)**

* JSON\_ARRAY\_APPEND: Thêm giá trị mới vào cuối mảng được chỉ định trong tài liệu JSON, trả về tài liệu JSON mới sau khi thêm, hoặc trả về NULL nếu bất kỳ đối số nào là NULL.

*Cú pháp:*

**JSON\_ARRAY\_APPEND(json\_doc, path, value[, path, value] ...)**

* JSON\_REPLACE: Cập nhật giá trị hiện có trong tài liệu JSON, trả về tài liệu JSON mới sau khi cập nhật, hoặc trả về NULL nếu bất kỳ đối số nào là NULL.

*Cú pháp:*

**JSON\_REPLACE(json\_doc, path, val[, path, val] ...)**

* JSON\_SET: Cập nhật hoặc thêm dữ liệu vào tài liệu JSON, trả về tài liệu JSON mới, hoặc trả về NULL nếu bất kỳ đối số nào là NULL hoặc nếu không tìm thấy đối tượng trong đường dẫn được cung cấp.

*Cú pháp:*

**JSON\_SET(json\_doc, path, val[, path, val] ...)**

* JSON\_REMOVE: Xóa dữ liệu khỏi tài liệu JSON, trả về tài liệu JSON mới sau khi xóa, hoặc trả về NULL nếu bất kỳ đối số nào là NULL. Nếu phần tử không tồn tại trong tài liệu thì không có thay đổi nào được thực hiện.

*Cú pháp:*

**JSON\_REMOVE(json\_doc, path[, path] ...)**

* JSON\_DETAILED (hoặc JSON\_PRETTY): Thể hiện JSON theo cách dễ hiểu nhất, nhấn mạnh các cấu trúc lồng nhau.

*Cú pháp:*

**JSON\_DETAILED(json\_doc[, tab\_size])**

*Hoặc*

**JSON\_PRETTY(json\_doc[, tab\_size])**

* JSON\_EXTRACT: Trích xuất dữ liệu từ tài liệu JSON dựa theo đường dẫn được cung cấp, hoặc trả về NULL nếu không có đường dẫn nào khớp hoặc nếu bất kỳ đối số nào là NULL.

*Cú pháp:*

**JSON\_EXTRACT(json\_doc, path[, path] ...)**

* JSON\_QUERY: Trả về một đối tượng hoặc mảng được chỉ định bởi đường dẫn, hoặc trả về NULL nếu tài liệu JSON không hợp lệ hoặc nếu không có kết quả phù hợp.

*Cú pháp:*

**JSON\_QUERY(json\_doc, path)**

* JSON\_VALUE: Trả về giá trị vô hướng được chỉ định bởi đường dẫn, hoặc trả về NULL nếu tài liệu JSON không hợp lệ hoặc nếu không có kết quả phù hợp.

*Cú pháp:*

**JSON\_VALUE(json\_doc, path)**

* JSON\_LENGTH: Trả về độ dài của tài liệu JSON, hoặc trả về độ dài của giá trị trong tài liệu được chỉ định bởi đường dẫn, hoặc trả về NULL nếu bất kỳ đối số nào là NULL hoặc nếu không xác định được giá trị trong đường dẫn.

*Cú pháp:*

**JSON\_LENGTH(json\_doc[, path])**

Độ dài sẽ được xác định như sau:

+ Độ dài của một giá trị vô hướng luôn là 1.

+ Nếu là mảng thì độ dài là số phần tử trong mảng.

+ Nếu là đối tượng thì độ dài là số lượng thành viên trong đối tượng.

* **Kịch bản demo:** Sử dụng các câu lệnh JSON làm việc trên tài liệu JSON có sẵn.
* **Link video demo:** [**https://drive.google.com/file/d/1FcpX5tCWebWQevi1p4\_wF35xaUpFbRqk/view?usp=sharing**](https://drive.google.com/file/d/1FcpX5tCWebWQevi1p4_wF35xaUpFbRqk/view?usp=sharing)

**CHƯƠNG 3: CÁC TÍNH NĂNG NÂNG CAO**

## **Phân quyền**

MariaDB cung cấp chức năng phân quyền truy cập nhằm đảm bảo dữ liệu được bảo mật và có thể kiểm soát quyền truy cập của các đối tượng người dùng để hạn chế các rủi ro liên quan đến cơ sở dữ liệu. Sau đây là một số câu lệnh cơ bản về phân quyền:

* CREATE USER: Tạo tài khoản user mới.

*Cú pháp:*

**CREATE USER [IF NOT EXISTS] user\_name** **IDENTIFIED BY 'password'**

*Hoặc*

**CREATE USER [IF NOT EXISTS] user\_name IDENTIFIED {VIA | WITH} authentication\_plugin {USING | AS} PASSWORD('password')**

* ALTER USER: Chỉnh sửa tài khoản user hiện có.

*Cú pháp:*

**ALTER USER [IF EXISTS] user\_name IDENTIFIED BY 'password'**

*Hoặc*

**ALTER USER [IF EXISTS] user\_name IDENTIFIED {VIA | WITH} authentication\_plugin {USING | AS} PASSWORD('password')**

* DROP USER: Xóa tài khoản user hiện có.

*Cú pháp:*

**DROP USER [IF EXISTS] user\_name**

* CREATE ROLE: Tạo role mới.

*Cú pháp:*

**CREATE ROLE [IF NOT EXISTS] role\_name [WITH ADMIN**

**{user\_name | role\_name}]**

* DROP ROLE: Xóa role hiện có.

*Cú pháp:*

**DROP ROLE [IF EXISTS] role\_name**

* GRANT: Cấp quyền cho tài khoản user, role.

*Cú pháp:*

**GRANT priv\_type ON [object\_type] priv\_level TO user\_name**

*Hoặc*

**GRANT role\_name TO user\_name, role\_name**

* REVOKE: Thu hồi quyền đã cấp cho tài khoản user, role.

*Cú pháp:*

**REVOKE priv\_type ON [object\_type] priv\_level FROM user\_name**

*Hoặc*

**REVOKE role\_name FROM user\_name, role\_name**

* **Kịch bản demo:**

+ Tạo 3 user mới sử dụng mật khẩu giống nhau.

+ Đăng nhập thử vào 1 user bất kỳ.

+ Đăng nhập lại vào root, đổi mật khẩu cho user đó.

+ Thử đăng nhập lại vào user đó với mật khẩu mới.

+ Đăng nhập lại vào root, xóa 1 user bất kỳ.

+ Tạo 3 role mới.

+ Xóa 1 role bất kỳ.

+ Cấp 2 role cho các user.

+ Thu hồi 1 role khỏi 1 user bất kỳ.

* **Link video demo:** [**https://drive.google.com/file/d/1221UZoqTjlQoN5YhRCxKWN0s09v8R-6Z/view?usp=sharing**](https://drive.google.com/file/d/1221UZoqTjlQoN5YhRCxKWN0s09v8R-6Z/view?usp=sharing)

## **Xác thực**

MariaDB cho phép thiết lập nhiều plugin xác thực cho một tài khoản người dùng. Khi tạo một tài khoản mới, nếu không chỉ định plugin xác thực cụ thể thì tài khoản vừa tạo mặc định sử dụng plugin mysql\_native\_password. Ngoài ra, MariaDB còn hỗ trợ các plugin khác như unix\_socket (dành cho Unix và Linux), ed25519,…

* mysql\_native\_password: Đăng nhập bằng tài khoản người dùng với mật khẩu tự thiết lập.

*Cú pháp:*

**mariadb -u user\_name -p**

* unix\_socket: Đăng nhập bằng tài khoản root@localhost (mặc định) hoặc bằng tài khoản khác mà không sử dụng mật khẩu thông qua Unix socket file cục bộ.

*Cú pháp:*

**mariadb**

* ed25519: Tương tự như mysql\_native\_password với cơ chế bảo mật tốt hơn.
* **Kịch bản demo:**

+ Tạo 1 user mới sử dụng plugin mysql\_native\_password và đăng nhập thử vào user đó.

+ Đăng nhập vào user root trên Linux.

* **Link video demo:** [**https://drive.google.com/file/d/1SML7BbRxPZrcBaT9oFde0bt83SwpKzbQ/view?usp=sharing**](https://drive.google.com/file/d/1SML7BbRxPZrcBaT9oFde0bt83SwpKzbQ/view?usp=sharing)

## **Backup và restore**

MariaDB hỗ trợ chức năng backup và restore thông qua hai câu lệnh mysqldump và mysql.

* Backup: Tạo bản sao lưu database trên ổ đĩa.

*Cú pháp:*

**mysqldump -u database\_user -p database\_name > backup\_file.sql**

* Restore: Sử dụng bản sao lưu database trên ổ đĩa để phục hồi.

*Cú pháp:*

**mysql -u database\_user -p database\_name < backup\_file.sql**

## **Import và export**

MariaDB hỗ trợ chức năng import và export thông qua hai câu lệnh LOAD DATA INFILE và SELECT INTO OUTFILE.

* Import: Nhập dữ liệu từ một file trên ổ đĩa.

*Cú pháp:*

**LOAD DATA INFILE file\_path**

**INTO TABLE table\_name**

**FIELDS TERMINATED BY ';'**

**LINES TERMINATED BY '\n';**

* Export: Xuất dữ liệu ra một file trên ổ đĩa.

*Cú pháp:*

**SELECT column 1, column 2,… FROM table\_name**

**INTO OUTFILE file\_path**

**FIELDS TERMINATED BY ';'**

**LINES TERMINATED BY '\n';**

* **Kịch bản demo:**

+ Backup một database vào thư mục bất kỳ với file backup tự đặt tên.

+ Sử dụng file backup đó restore vào database khác.

+ Tạo bảng mới và import dữ liệu vào bảng từ file csv.

+ Export dữ liệu bảng ra file csv lưu vào thư mục bất kỳ.

* **Link video demo:** [**https://drive.google.com/file/d/1PltdtHihV1BObTPDXFyvmEv8d6r6\_UXk/view?usp=sharing**](https://drive.google.com/file/d/1PltdtHihV1BObTPDXFyvmEv8d6r6_UXk/view?usp=sharing)

**CHƯƠNG 4: SO SÁNH MARIADB VỚI MYSQL**

|  |  |  |
| --- | --- | --- |
|  | **MariaDB** | **MySQL** |
| **Tốc độ và hiệu suất** | Nhanh và tối ưu hơn MySQL | Chậm hơn MariaDB |
| **Đối tượng database** | Theo tiêu chuẩn SQL, hỗ trợ đối tượng sequence từ phiên bản 10.3 | Theo tiêu chuẩn SQL, nhưng không hỗ trợ đối tượng sequence |
| **JSON** | Tài liệu được lưu trữ dưới dạng chuỗi | Tài liệu được lưu trữ dưới dạng đối tượng nhị phân |
| **Khả năng tương thích với Oracle** | Mức độ tương thích cao và hỗ trợ PL/SQL từ phiên bản 10.3 | Mức độ tương thích cao, nhưng không hỗ trợ PL/SQL |
| **Mã hóa** | Mã hóa bản ghi tạm thời và bản ghi nhị phân | Mã hóa dữ liệu đang được lưu trữ với InnoDB và thuật toán AES |
| **Xác thực** | Chỉ hỗ trợ xác thực thông qua các plugin xác thực | Đi đầu về bảo mật, triển khai xác thực với thuật toán SHA-256 |
| **Storage engine** | Hỗ trợ nhiều storage engine hơn so với MySQL | Hỗ trợ ít storage engine hơn so với MariaDB |
| **Giấy phép** | Theo GPL | Có hai phiên bản là Enterprise và GPL |
| **Thread pool** | Hỗ trợ miễn phí hơn 200.000 kết nối | Chỉ hỗ trợ hơn 200.000 kết nối ở bản Enterprise (trả phí) |

**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **Kết luận**

MariaDB là một hệ quản trị cơ sở dữ liệu mạnh mẽ, tiềm năng, được tin tưởng và sử dụng rộng rãi trong cộng đồng phát triển phần mềm mã nguồn mở. Nhờ việc được hỗ trợ liên tục từ cộng đồng, MariaDB luôn luôn được cải tiến nhanh chóng và phù hợp với nhu cầu thực tiễn của người dùng. Thêm vào đó, vì là nhánh phát triển riêng biệt của MySQL, MariaDB thừa hưởng đầy đủ các tính năng ổn định của MySQL và ngoài ra có khả năng mở rộng cao, đem lại sự tin cậy và độ hiệu quả cao trong việc quản lý và bảo vệ dữ liệu.

## **Hướng phát triển**

Với những ưu điểm vượt trội so với MySQL và các cơ sở dữ liệu khác, MariaDB mang lại nhiều lợi thế lớn cho các công ty, tập đoàn, doanh nghiệp dự định phát triển các dự án khả thi và tiềm năng từ quy mô nhỏ đến lớn. MariaDB có thể được áp dụng trong nhiều lĩnh vực khác nhau, bao gồm:

* Data warehouse (Kho dữ liệu): MariaDB có khả năng tích hợp, lưu trữ dữ liệu đa nguồn, xử lý các tác vụ phân tích và truy vấn dữ liệu phức tạp với hiệu suất cao.
* Web application (Ứng dụng Web): MariaDB lưu trữ dữ liệu một cách hệ thống, cho phép nhiều người dùng đồng thời truy xuất thông tin và trả về kết quả mong muốn một cách nhanh chóng trên trang web. Cụ thể, MariaDB có thể được sử dụng để phát triển các trang thương mại điện tử đáp ứng nhu cầu của khách hàng trong việc tìm kiếm sản phẩm.
* Mobile application (Ứng dụng di động): MariaDB kết hợp với các framework để xây dựng ứng dụng di động đa nền tảng, cung cấp khả năng đồng bộ và xử lý dữ liệu linh hoạt với tính bảo mật cao.
* Hệ thống IoT (Internet of Things): MariaDB có thể lưu trữ dữ liệu được ghi nhận từ các thiết bị IoT bao gồm các giá trị đo lường, thời gian thực, trạng thái,… để phân tích, thống kê và báo cáo.
* Machine learning (Máy học): MariaDB kết hợp với MindsDB có thể huấn luyện các mô hình máy học thông qua SQL, cho phép dự đoán và phân tích dữ liệu.

**TÀI LIỆU THAM KHẢO**

[1] “MariaDB Official Website.” https://mariadb.com

[2] “MariaDB Documentation.” https://mariadb.org/documentation

[3] “What Is MariaDB?” https://www.purestorage.com/nl/knowledge/what-is-mariadb.html

[4] “What is MariaDB? A comparison with MySQL.”

https://www.opc-router.com/what-is-mariadb-a-comparison-with-mysql

[5] “What’s the Difference Between MariaDB and MySQL?” https://aws.amazon.com/compare/the-difference-between-mariadb-vs-mysql

[6] dbForge Team, “MySQL vs MariaDB: Difference, Performance, and Features (Full Comparison),” 2022. https://blog.devart.com/mysql-vs-mariadb.html

[7] “Thiết kế hệ thống IoT đơn giản dùng giao thức MQTT kết hợp PHP-MySQL và Node.js,” 2020. https://tapit.vn/thiet-ke-thong-iot-don-gian-dung-giao-thuc-mqtt-ket-hop-php-mysql-va-node-js

[8] “Machine Learning inside MariaDB.” https://mindsdb.com/mariadb-machine-learning