**DATABASE TESTING**

*Phạm Đình Thiện, Trần Văn Thắng, Phùng Văn Dương,*

*Cao Văn Cường, Hồ Nam Anh, Trần Hải Dương*

**TÓM TẮT**

Cơ sở dữ liệu là một trong những phần tất yếu của một ứng dụng phần mềm. Khi độ phức tạp của một ứng dụng tăng lên, nhu cầu về một Cơ sở dữ liệu mạnh mẽ và an toàn xuất hiện. Để đảm bảo các dữ liệu đầu vào có được lưu trữ một cách hợp lệ trong Cơ sở dữ liệu và tránh việc mất mát dữ liệu hay việc truy cập trái phép vào Cơ sở dữ liệu nhóm 12 đưa ra một giải pháp giúp giải quyết những vấn đề trên đó là xây dựng một công cụ kiểm thử Cơ sở dữ liệu giúp lập trình viên giải quyết những vấn đề trên đặc biệt là hai phần Convention và SQL Injection.

**Từ khóa: Database Testing, Convention, SQL Injection.**

**I. Hiện trạng**

Thông qua việc khảo sát 7 khối trong Khối Sản Xuất bao gồm Khối DA nền tảng quản trị kinh doanh, Khối DA Hành chính sự nghiệp, Viện Đào tạo và Nghiên cứu Công nghệ, Khối DA nền tảng Quản trị Nhân sự - Điều hành, Khối DA nền tảng tài chính DN, Khối DA nền tảng giáo dục, Khối Customize. Đội thấy rằng hiện nay các khối đang chưa có sẵn các công cụ liên quan đến việc kiểm tra convention database, SQL Injection. Vì vậy, đây chính là cơ hội tốt để đội tìm hiểu, nghiên cứu và đưa ra ý tưởng mới về việc xây dựng giải pháp giúp giải quyết các vấn đề liên quan đến kiểm tra convention database, cũng như SQL Injection, nhằm hỗ trợ, giúp tăng hiệu quả, năng suất cho các dự án.

**Về Convention**

Theo khảo sát mỗi dự án chiếm tỷ lệ từ 10% đến 20% database chứa các tên bảng, tên store, tên trường, … chưa đúng theo Convention chung, điều đó dẫn đến khi triển khai trên các môi trường thường có sự sai sót như mang nhầm bảng, nhầm store,... dẫn đến lỗi chương trình. Đặc biệt là các database được nhiều sản phẩm dùng chung thì việc kiểm soát và triển khai trở nên khó khăn hơn. Ngoài ra, việc không tuân thủ Convention còn dẫn tới việc thi công trở nên phức tạp, chậm trễ do có thể mapping tên trường sai, viết các câu truy vấn không chính xác do nhầm lẫn hoa thường dẫn đến việc tìm lỗi và sửa mất nhiều thời gian. Các lập trình viên mới khi tiếp nhận công việc cảm thấy khó khăn hơn khi việc tìm hiểu cấu trúc database mất nhiều thời gian do cấu trúc không thống nhất các phần dễ gây nhầm lẫn.

Nguyên nhân dẫn đến việc lập trình viên chưa tuân thủ theo đúng Convention có nhiều nguyên nhân. Những lập trình mới khi vào dự án chưa được đào tạo kỹ về việc tuân thủ theo đúng quy trình quy định khi thi công ở database. Việc chuyển đổi dự án giữa các lập trình viên cũng là nguyên nhân khi lập trình viên vào dự án mới chưa biết được convention của dự án như thế nào.

Ví dụ: Tại dự án C&B có 3 sản phẩm Chấm Công, Hồ Sơ, Tiền Lương cùng sử dụng chung một database với sản phẩm khác là MinTax, khi có một lập trình viên mới vào dự án chưa biết convention của dự án đã đặt tên bảng không chứa ký tự bắt đầu là “ts\_” dẫn đến khi triển khai DevOps không biết bảng dữ liệu đó là của sản phẩm mình nên đã không mang ra ngoài môi trường Internet dẫn đến lỗi và phải thực hiện triển khai lại mất rất nhiều thời gian.

**Về SQL Injection**

SQL injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject và thi hành các câu lệnh SQL bất hợp pháp. Đây là lỗ hổng thường gặp nhất của các website trên internet và nằm trong 10 lỗ hổng hàng đầu của OWASP . Lỗ hổng này nằm trong các ứng dụng web chứ không phải vấn đề ở cơ sở dữ liệu hay web server. Hầu hết các lỗi này xuất phát từ lập trình viên.

Hiện nay tại MISA SQL Injection được kiểm tra qua ban an ninh mà chưa có công cụ kiểm tra dành riêng cho lập trình viên trong khi thi công. Việc kiểm tra SQL Injection khi thi công đang thực hiện một cách thủ công bằng cách nhờ những người có kinh nghiệm như các SA kiểm tra và đánh pass SQL Injection bằng cách thêm comment để đánh dấu đã hoàn thành kiểm tra. Việc này sẽ rất tốn thời gian và độ chính xác cũng tùy thuộc vào khả năng của người review, khiến cho lỗi SQL Injection có thể bị lọt ra ngoài.

**II. Nguyên nhân và hậu quả**

**Nguyên nhân**

* DEV mới vào chưa rõ convention.
* DEV cũ chưa nắm được cách đặt tên theo convention.
* Chưa có công cụ check SQL injection cho DEV.

**Hậu quả**

* Tìm kiếm tên bảng, thủ tục, … mất thời gian do không đồng nhất.
* Bên ban An ninh kiểm tra an ninh phát hành, nếu có phát hiện lỗi lọt, dẫn đến dự án sẽ mất thời gian sửa lại.
* Bị hacker tấn công hệ thống do SQL Injection.

**III. Khái niệm**

**Database Testing**

Database Testing – Kiểm tra cơ sở dữ liệu là một loại kiểm thử phần mềm nhằm kiểm tra các bảng, lược đồ của Cơ sở dữ liệu (Database). Nó cũng kiểm tra tính toàn vẹn và nhất quán của dữ liệu, kiểm tra hoạt động của Cơ sở dữ liệu trước những luồng dữ liệu đầu vào bất thường.

**Convention**

Convention là một phần rất quan trọng trong việc thi công và giám sát. Nó giúp cho các lập trình viên có thể dễ dàng kiểm soát dữ liệu của dự án hay bảo trì, phát triển các tính năng mới. Convention càng chặt chẽ thì hiệu quả và chất lượng càng được tăng cao. Bản thân MISA cũng đã có quy trình quy định về việc tuân thủ Convention và trong đó Database cũng chiếm một phần rất quan trọng. Tuy nhiên việc tuân thủ và làm theo quy định của lập trình viên chưa được giám sát chặt chẽ dẫn đến có nhiều dự án vẫn chưa đảm bảo Database của dự tuân theo đúng Convention mà công ty đã đặt ra.

**SQL Injection**

SQL injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp. SQL injection có thể cho phép những kẻ tấn công thực hiện các thao tác, delete, insert, update, v.v. trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy. SQL injection thường được biết đến như là một vật trung gian tấn công trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase… (Theo wiki).

**IV. Giải pháp đề xuất**

Tạo một tool hỗ trợ DEV trong việc thi công trên Database với các chức năng:

* Kiểm tra Convention.
* Kiểm tra SQL Injection.

Giải pháp đưa ra được dựa hoàn toàn theo kinh nghiệm của các thành viên trong đội và nhu cầu mong muốn của thành viên trong dự án.

Ưu điểm: Hỗ trợ, tối ưu thời gian, tăng hiệu quả, năng suất cho việc thi công database.

Nhược điểm:

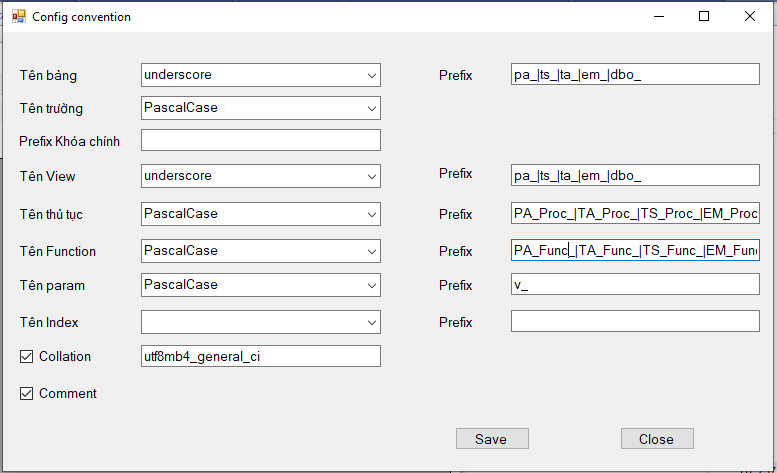
* DEV cần thời gian làm quen tool và có thể làm ảnh hướng đến thói quen code.
* Tool chỉ áp dụng cho án sử dụng SQL (MySql).

Cách đo đếm hiệu quả: Dựa trên thời gian thi công, thời gian sửa lỗi Khách hàng.

**V. Giải pháp chi tiết**

**1. Kiểm tra convention**  
**Naming Convention**

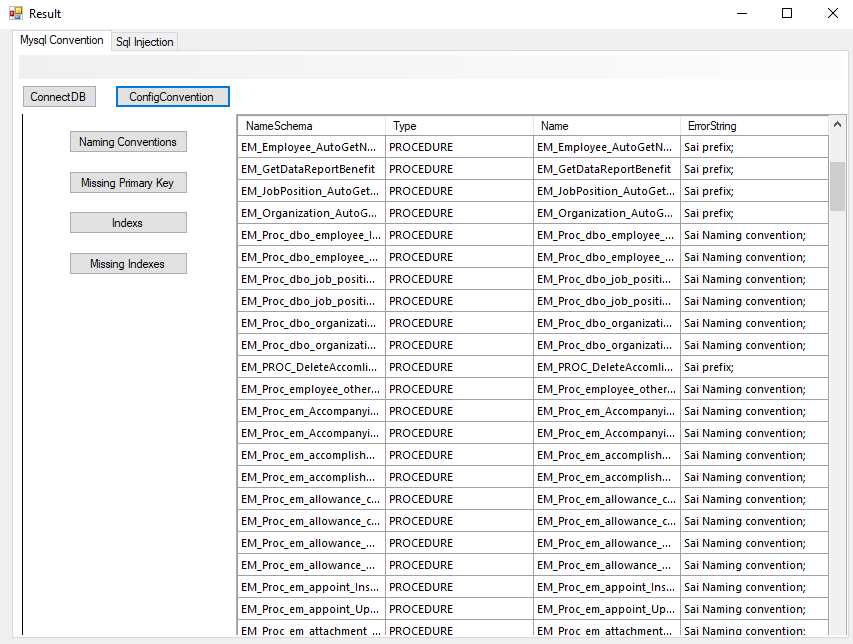
Tool Database Testing cho phép người dùng thiết lập các quy tắc đặt tên bảng, trường dữ liệu, khóa chính, view, thủ tục, function, param, index,... có comment đầy đủ các phần hay không. Ví dụ đối với tên bảng có thể thiết lập Underscore, Camel Case, Pascal Case và ký tự bắt đầu phải thỏa mãn một trong các từ sau pa\_|ts\_|ta\_|em\_|dbo\_ (các giá trị được phân cách bởi dấu ‘|’).



*Thiết lập các quy tắc đặt tên theo convention*

Sau đó dựa vào các câu lệnh xuất thông tin cấu hình của database(information\_schema.TABLES,information\_schema.COLUMNS,information\_schema.STATISTICS,information\_schema.PARAMETERS) lấy ra thông tin đầy đủ cấu hình của database để kiểm tra theo các quy tắc đã thiết lập ở config và hiện lên danh sách các phần chưa tuân thủ theo đúng quy tắc đã thiết lập.

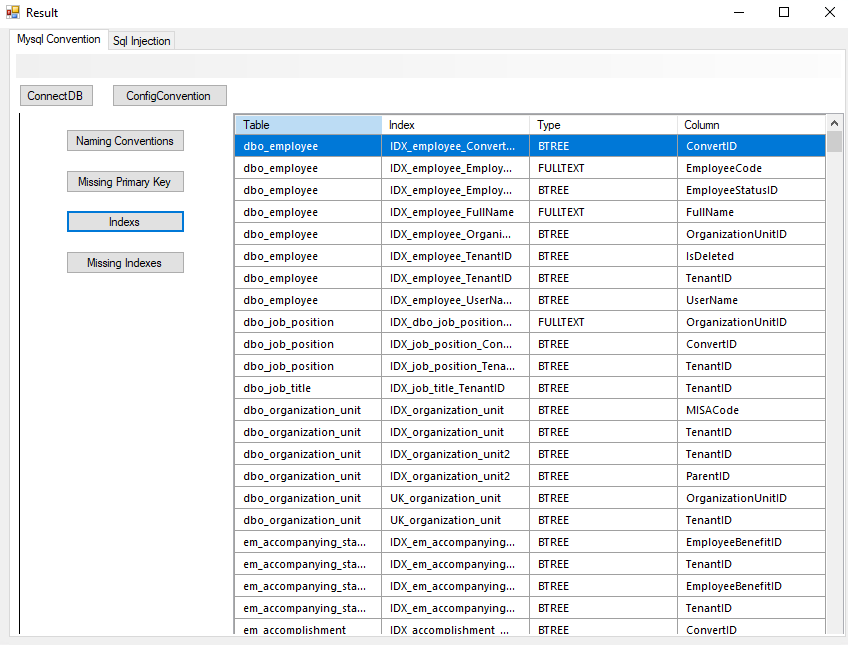
Dựa vào kết quả trả về trưởng nhóm hoặc người giám sát có thể dễ dàng biết được những phần nào chưa tuân theo đúng quy tắc convention mà dự án đã đặt ra một cách nhanh chóng và kịp thời sửa đổi mà không cần vào database đi kiểm tra từng phần một mất rất nhiều thời gian và có khả năng sai sót lớn.



*Danh sách kết quả trả về sau khi kiểm tra*

**Danh sách Index**

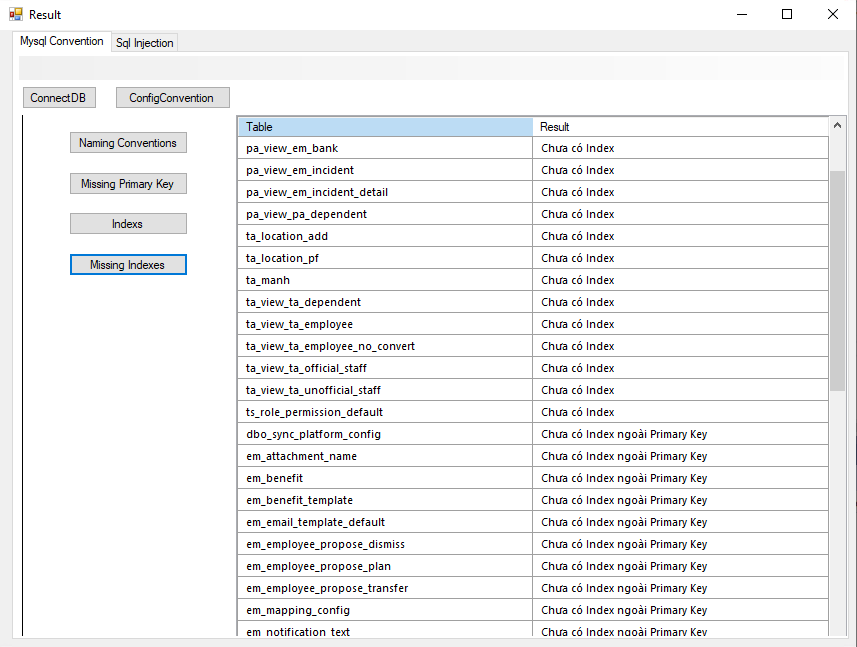
Việc đánh index cho các bảng dữ liệu rất quan trọng vì đó là một phần giúp việc truy xuất dữ liệu có hiệu quả tốt hơn. Dựa vào việc truy vấn dữ liệu từ information\_schema.STATISTICS tool Database Testing hiển thị thông tin đầy đủ các index của từng bảng dữ liệu để người dùng có thể dễ dàng giám sát. Người dùng có thể tìm kiếm các bảng mà mình muốn kiểm tra để biết được cách thức sử dụng index của bảng dữ liệu đó.

**

*Danh sách index được sử dụng ở database*

**Chưa được đánh index**

Ngoài việc hiển thị tất cả các index được sử dụng trên database tool Database Testing còn cho phép người dùng truy vấn những bảng dữ liệu chưa được đánh được đánh bất kỳ index nào và chưa có index nào khác ngoài khóa chính.

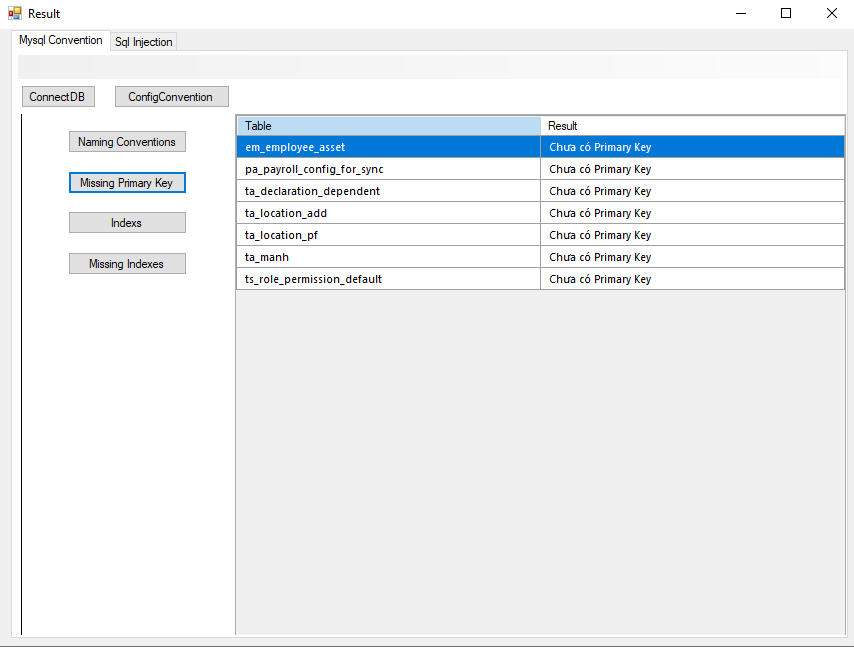


*Danh sách các bảng dữ liệu chưa được đánh index*

Khi có một phân hệ có lượng dữ liệu rất lớn và có thời gian truy xuất dữ liệu rất chậm ta có thể kiểm tra liệu bảng dữ liệu này đã được sử dụng index hay chưa. Việc biết được những bảng dữ liệu nào chưa được sử dụng index giúp cho việc hiệu năng chương trình trở nên đơn giản và nhanh chóng hơn.

**Chưa sử dụng khóa chính**

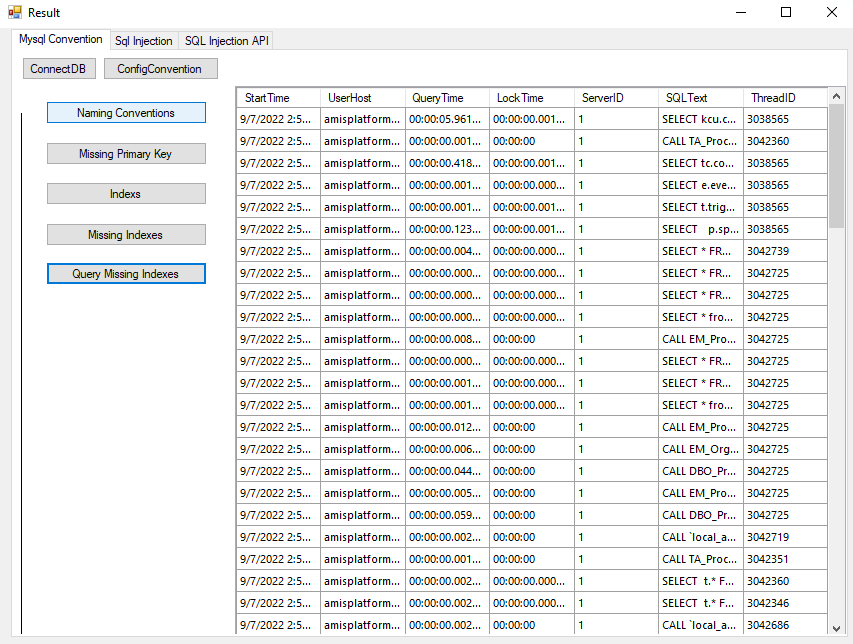
Khóa chính là một phần rất quan trọng đối với bảng dữ liệu trong database. Khóa chính vừa có tác dụng định danh từng bản ghi và vừa có tác dụng tối ưu khả năng truy vấn dữ liệu (giống như một index). Nên việc cả bảng dữ liệu không có khóa chính cũng là một vấn đề khiến hệ thống có hiệu quả chưa tốt. Tool Database Testing cho phép người dùng truy vấn các bảng dữ liệu trong database chưa được thiết lập khóa chính. Người dùng có thể dễ dàng nắm bắt được việc có bảng dữ liệu nào quan trọng mà chưa được thiết lập khóa chính hay không.



*Danh sách các bảng chưa thiết lập Primary Key*

**Các câu truy vấn chưa sử dụng index**

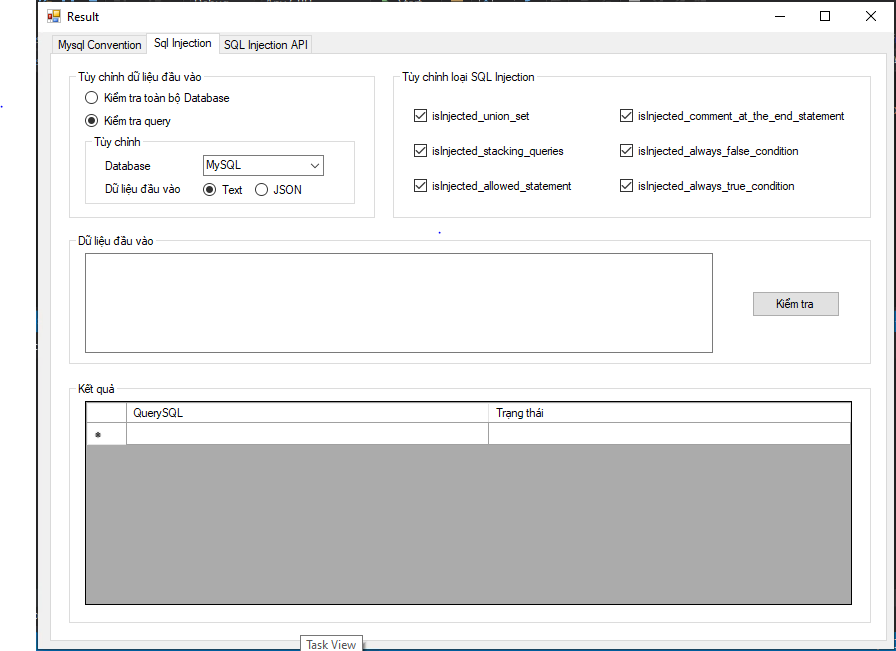
Các lệnh truy vấn đóng vai trò quan trọng là cầu nối trung gian trong việc truyền dữ liệu qua lại từ phía service tới database, câu truy vấn càng tối ưu thì thời gian thực thi chức năng càng nhanh chóng và hiệu quả. Tool database testing cho phép người dùng truy vấn tất cả các truy vấn trong quá trình người dùng sử dụng mà chưa được sử dụng bất cứ trường dữ liệu được đánh index ở trong mệnh đề điều kiện. Người dùng có thể dễ dàng theo dõi các lệnh truy vấn được sử dụng có thời gian truy vấn cao, tần suất được thực thi giúp cho việc phát triển hệ thống, giám sát bảo trì hệ thống trở nên hiệu quả hơn.

****

*Danh sách các truy vấn đã được thực thi nhưng chưa sử dụng index*

**2. Kiểm tra SQL Injection**

Tool chứa 2 tab tính năng liên quan đến kiểm thử SQL Injection. Tính năng thứ nhất được sử dụng để kiểm tra SQL Injection cho dữ liệu đầu vào dưới dạng text. Người dùng có thể truyền một câu lệnh cụ thể hoặc truyền nhiều câu lệnh dưới dạng JSON bằng cách thay đổi trạng thái radio Dữ liệu đầu vào.



*UI 2 tab chính để kiểm tra sql injection.*

Việc tìm kiếm tất cả các loại SQL Injection là điều không thể. Tuy nhiên những trường hợp được liệt kê dưới đây có thể giúp một dự án bao phủ được phần lớn khả năng SQL Injection thường xuyên xảy ra và các trường hợp này cũng đã được tích hợp trong việc kiểm thử của tool:

* Câu lệnh where bị chèn thêm các dòng như ‘a’ = ‘a’ hoặc 1 = 1.

VD: select col1 from table1 where col1 > 1 or 1=1;

* Câu lệnh where bị rút ngắn bằng cách thêm comment (--).

VD: select col1 from table1 where col1 > 1 -- and col2 > 2;

* Câu lệnh chứa UNION, có khả năng đọc được các thông tin của bảng khác hoặc thông tin database.

VD: select col1 from table1 where col1 > 1 union select col2 from table2;

* Có nhiều hơn một câu lệnh trong một transaction.

VD: select col1 from table1 where col1 > 1; update table2 set col1=1;

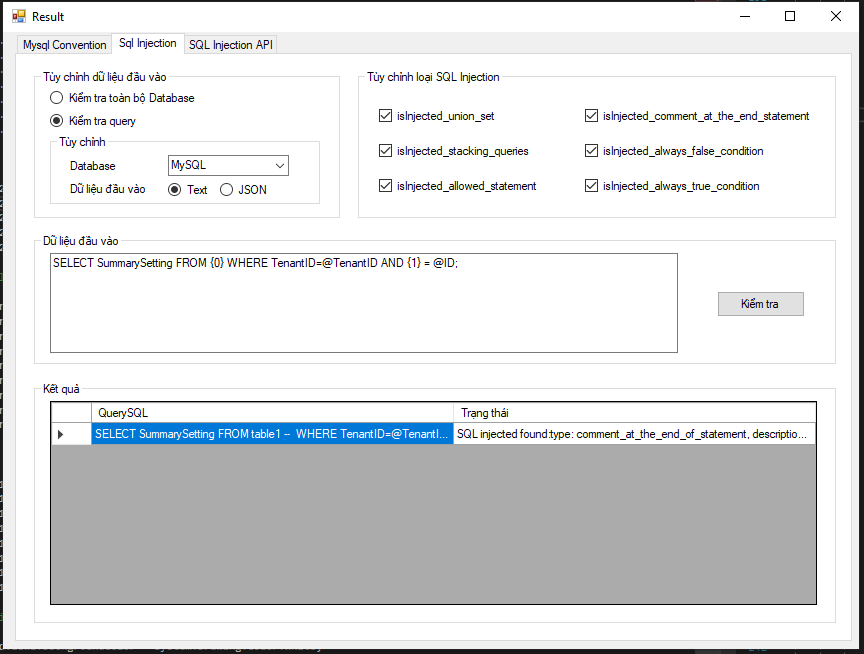
* Câu lệnh SQL bị chèn thêm các dòng SQL không xác định trước.

VD: select col1 from table1 where col1 > 1; drop table t1;

* Câu lệnh được chèn thêm việc gọi các thủ tục.

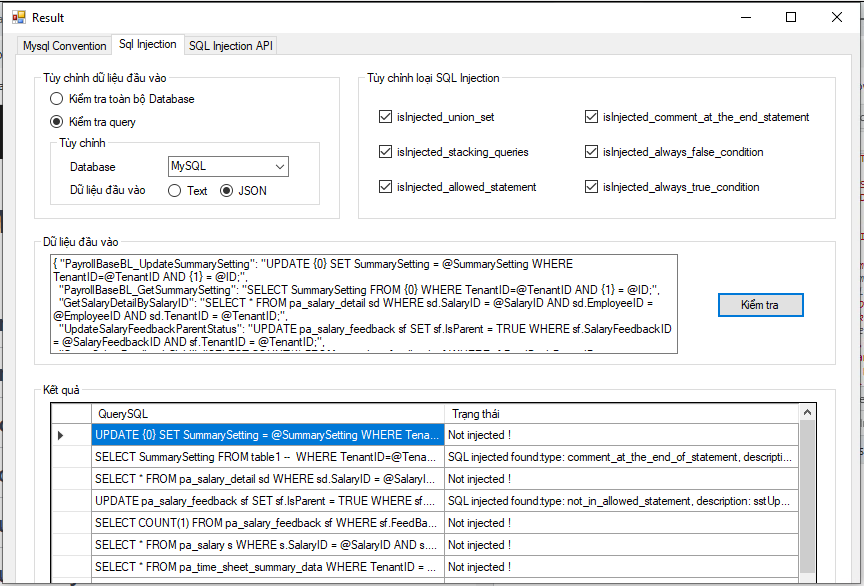
VD: select col1 from table1 where col1 > 1; CALL store\_insert(val1, val2);

Để thực hiện kiểm tra SQL Injection dạng Text, người dùng chọn mode Text và nhập câu lệnh SQL vào ô input Dữ liệu đầu vào, sau đó nhấn button Kiểm tra. Tool sẽ thực hiện và đưa ra kết quả.



*UI kiểm tra sql injection cho dữ liệu đầu vào dạng text (1 câu lệnh sql)*

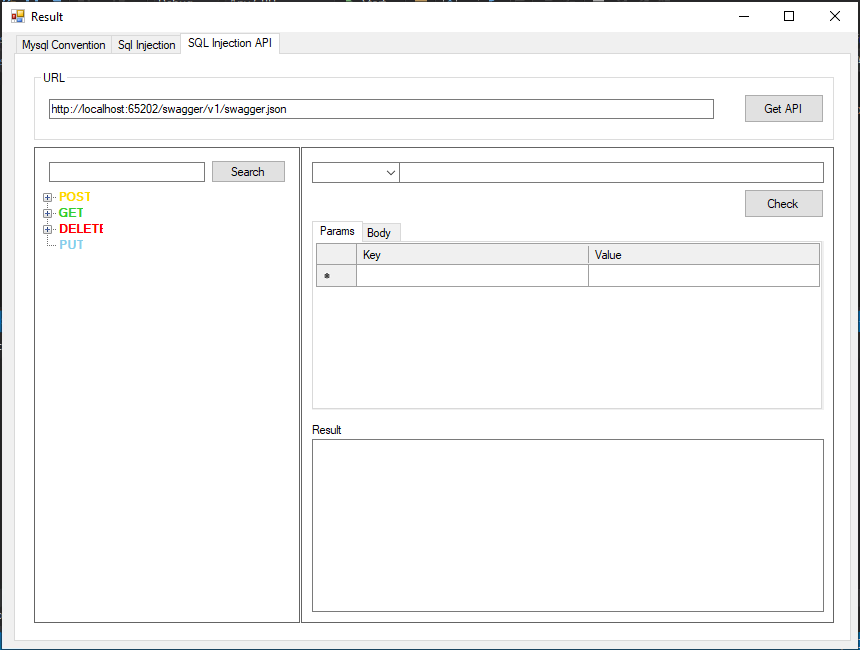
Để thực hiện kiểm tra SQL Injection nhiều câu lệnh SQL hơn trong một thời điểm, người dùng chọn mode JSON và nhập các câu lệnh SQL vào ô input Dữ liệu đầu vào, sau đó nhấn button Kiểm tra. Tool sẽ thực hiện và xuất ra kết quả đối với từng câu lệnh.



*UI kiểm tra sql injection cho dữ liệu đầu vào dạng JSON (nhiều câu lệnh sql)*

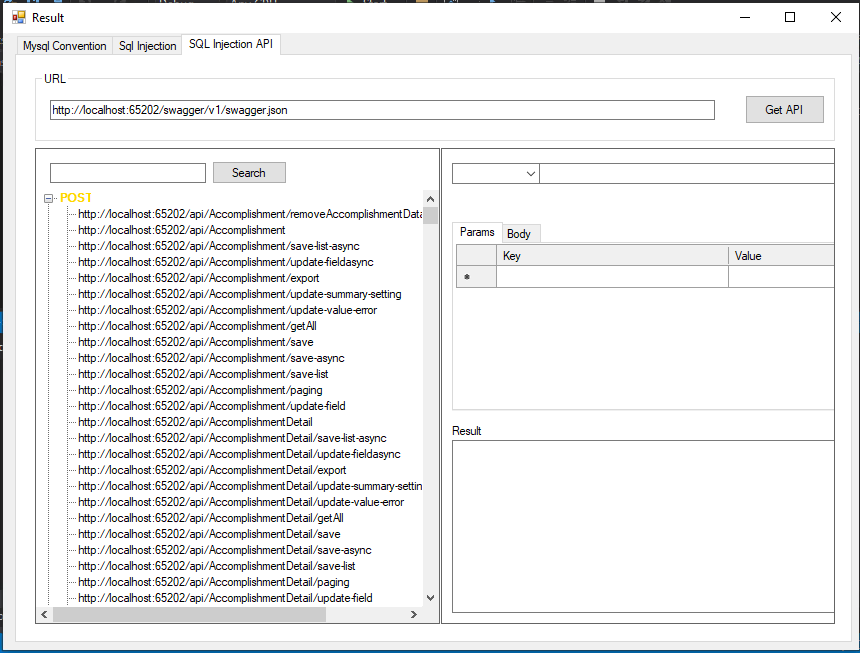
Tuy nhiên, với tính năng thứ nhất, tool chỉ đáp ứng cho việc kiểm tra các dòng lệnh SQL cụ thể. Vì vậy, để việc kiểm thử SQL Injection đạt được hiệu quả cao nhất trong một dự án, chúng ta cần kiểm tra rò rỉ thông qua việc gọi API. Do đó, đội phát triền đã xây dựng tính năng thứ hai, giúp tìm kiếm rủi ro SQL Injection chính xác hơn thông qua việc thực thi một API cụ thể, và giả lập các tham số truyền vào API có dạng text, bằng cách thay thế các tham số đó thành các chuỗi ký tự có khả năng gây ra SQL Injection. Ngoài ra, tool cũng cung cấp khả năng liệt kê các đầu API của dự án thông qua swagger để người sử dụng tiện thao tác.

Để hiển thị toàn bộ các đầu API từ swagger, người dùng nhập đường dẫn URL swagger, sau đó bấm button Get API. Tool sẽ thực hiện quét và hiện ra toàn bộ API gom nhóm theo Methods (POST, GET, DELETE, PUT).



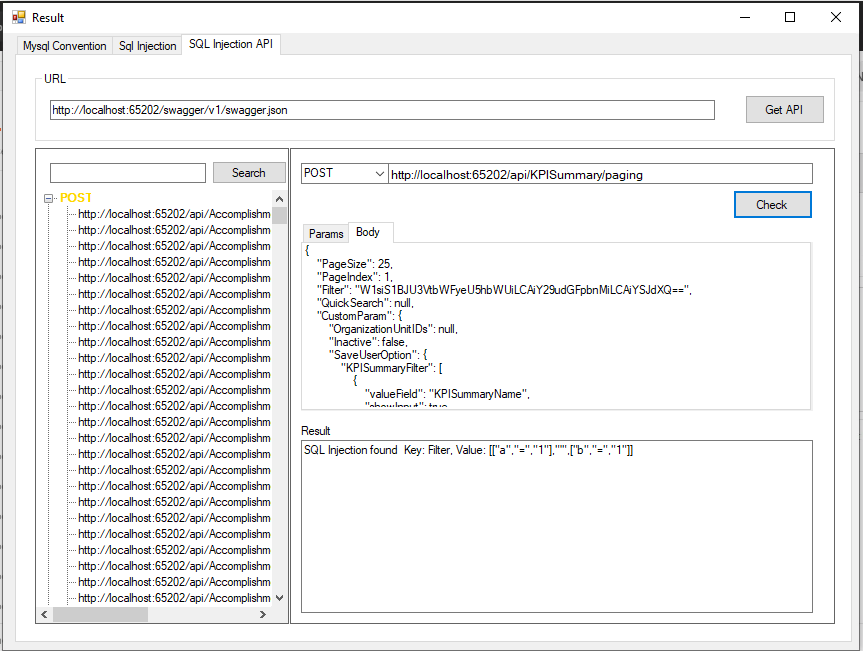
*UI liệt kê tất cả các đầu API qua swagger*

Việc hiển thị các đầu API gom nhóm theo methods sẽ giúp người dùng thuận tiện trong việc thao tác hơn.



*UI liệt kê tất cả các đầu API qua swagger*

Để thực hiện kiểm tra SQL Injection cho một đầu API cụ thể, người dùng click chuột vào đường dẫn API mong muốn, nhập dữ liệu cho header và body, sau đó bấm button Check để tool bắt đầu thực hiện kiểm tra. Sau khi thực hiện xong, tool sẽ trả ra kết quả thông báo Key nào trong body đang gây ra SQL Injection và Value tương ứng kèm theo.



*UI kiểm tra SQL Injection bằng cách gọi API cụ thể và xuất ra kết quả Injection*

**VI. Kế hoạch phát triển**

Tại phiên bản đầu tiên - phiên bản thử nghiệm Tool Database Testing Alpha 1.0.0 đã cung cấp hai tính năng quan trọng nhất là MySQL Convention và SQL Injection, hai phần này cũng chính là điểm cốt lõi trong quá trình phát triển tiếp theo của đội ngũ sản xuất. Trong thời gian tới, đội ngũ phát triển sẽ tiếp tục tập trung hoàn thiện hai tính năng chính phía trên. Cụ thể các cải tiến đang được đội nghiên cứu và xây dựng : truy vấn những nơi các trường dữ liệu được sử dụng trên database (view, procedure, trigger, …), kiểm tra SQL Injection API các API DELETE, UPDATE dữ liệu, … Sau đó, ngoài hai phần cốt lõi chính, đội ngũ phát triển sẽ có những tính năng mới được tích hợp vào tool như: tự động tạo model từ bảng ở database thành dạng js/c#/ts, ghi log query param trên môi trường chính thức để hỗ trợ kiểm tra lỗi, . . . nhằm hỗ trợ tối đa việc phát triển, giám sát và bảo trì hệ thống.

**Lộ trình phát triển chi tiết:**

* Alpha 1.0.0
* Kết nối với database cần kiểm tra.
* Thiết lập các quy tắc convention của dự án.
* Kiểm tra naming convention dựa vào thiết lập quy tắc của dự án.
* Truy xuất các bảng chưa cài đặt khóa chính.
* Truy xuất danh sách tất cả index được sử dụng.
* Truy xuất các bảng chưa sử dụng index.
* Kiểm tra SQL Injection các câu lệnh MySQL theo chuẩn 6 loại phổ biến nhất hiện nay.
* Kiểm tra SQL Injection các đầu API lấy dữ liệu paging.
* Alpha 2.0.0
* Kiểm tra SQL Injection các đầu API cập nhật và xóa dữ liệu.
* Tự động tạo model từ bảng database.
* Version 1.0.0
* Gen model từ bảng database.
* Ghi log query param khi chạy các procedure trên các môi trường.

**Kế hoạch phát hành:**

| Phiên bản | Ngày phát hành |
| --- | --- |
| Alpha 1.0.0 | 05/09/2022 |
| Alpha 2.0.0 | 19/09/2022 |
| Version 1.0.0 | 03/10/2022 |

Dự kiến các phiên bản phát hành cách nhau mỗi tháng nhằm cải tiến các tính năng cũ và tích hợp thêm nhiều tính năng mới khác liên quan tới việc nâng cao chất lượng phát triển và giám sát database trong quá trình xây dựng sản phẩm.

**VII. Tổng kết**

Nhìn chung Tool Database Testing là công cụ đắc lực để hỗ trợ các lập trình viên trong việc giám sát chất lượng và hiệu năng của database (MySQL/SQL), trong quá trình xây dựng phát triển sản phẩm cũng như trong quá trình bảo trì sản phẩm, giúp cho việc kiểm tra rà soát tổng thể database về nhiều khía cạnh một cách dễ dàng nhanh chóng hơn.

Đội ngũ phát triển tin rằng việc áp dụng Tool Database Testing vào trong công việc sẽ mang lại hiệu quả cao và nhanh chóng cho các đội ngũ lập trình viên.