|  |
| --- |
| **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  **KHOA CÔNG NGHỆ THÔNG TIN 2**  🙞🙞🙞🕮🙜🙜🙜    **BÁO CÁO CHUYÊN ĐỀ KHOA HỌC**  **CẤP HỌC VIỆN**  **CÁC CƠ CHẾ ĐẢM BẢO AN TOÀN DỮ LIỆU**  **TRONG SQL SERVER**  **Báo cáo viên: ThS. Lưu Nguyễn Kỳ Thư**  **Năm học: 2021-2022**  *Thành phố Hồ Chí Minh, tháng 10/2022* |

**Phần 1: Giới thiệu mở đầu**

Hệ quản trị cơ sở dữ liệu SQL Server là 1 phần mềm giúp cho chúng ta có thể lưu giữ database, hỗ trợ nhiều người dùng có thể dùng chung 1 cơ sở dữ liệu.

Chuyên đề nhằm mục đích tìm hiểu các khả năng dữ liệu không an toàn khi có nhiều người dùng sử dụng chung 1 cơ sở dữ liệu, và đưa ra các phương pháp khắc phục.

**Mục tiêu của chuyên đề:**

* Tìm hiểu các phương pháp đảm bảo an toàn dữ liệu trong SQl Server
* Nội dung:   
  ⮱ Tìm hiểu các khả năng dữ liệu không an toàn khi có nhiều người dùng sử dụng chung 1 cơ sở dữ liệu

⮱ Các phương pháp đảm bảo an toàn dữ liệu trong SQl Server

**Phần 2: Nội dung chuyên đề khoa học**

### **Các mối đe dọa bảo mật trong SQL Server**

Dưới đây là các mối đe dọa bảo mật phổ biến ảnh hưởng đến cơ sở dữ liệu trong SQL Server:

* Xác thực vào SQL — Đăng nhập máy chủ SQL có thể dễ dàng bị tấn công bằng cách chèn vào chuỗi kết nối 1 số ký tự đặc biệt làm vô hiệu hóa password. Khi một chuỗi kết nối được xây dựng tại thời điểm đăng nhập, kẻ tấn công có thể thêm các ký tự đặc biệt để có thể thực hiện các hành động trái phép trên máy chủ.
* SQL injection — SQL Server, giống như các cơ sở dữ liệu khác, dễ bị tấn công trong đó người dùng chèn thêm lệnh vào chuỗi truy vấn. Các lệnh này có thể làm hỏng cơ sở dữ liệu hoặc được sử dụng để lọc dữ liệu nhạy cảm. Để ngăn chặn việc đưa thêm lệnh vào SQL, ta kiểm tra tất cả các input để đảm bảo chúng không chứa các ký tự có thể được sử dụng để thực thi mã.

### **SQL Server Security Configuration**

Microsoft SQL Server cung cấp các mức bảo mật: phân quyền truy xuất, auditing, và mã hóa.

**2.1. Phân quyền Truy xuất**

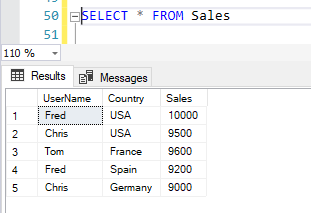
* Quyền truy cập dựa trên Role: SQL Server cho phép kiểm soát quyền truy cập vào dữ liệu ở ba mức: Server, cơ sở dữ liệu và bảng.
* Bảo mật mức hàng (**Row-Level Security -** RLS) và Bảo mật mức cột (**Row-Level Security -** CLS): ta có thể xác định quyền truy cập có điều kiện vào các hàng hoặc cột cụ thể trong table. Đây là một công cụ mạnh mẽ để bảo vệ dữ liệu nhạy cảm hoặc các đối tượng dữ liệu theo các yêu cầu cần tuân thủ.

Các bước cài đặt Row-Level Security trong SQL Server

1. Ta tạo các users muốn truy xuất dữ liệu. Các users này sẽ được cấp quyền truy cập vào một số bản ghi dựa trên ngữ cảnh đăng nhập của họ.
2. Bước tiếp theo là tạo một **inline table-valued function** trong SQL. Hàm này sẽ chứa bộ lọc cho bảng mà RLS sẽ được triển khai
3. Bước cuối, ta tạo **security policy** cho table và cung cấp inline table-valued function ở trên cho table.

Ví dụ: Ta sẽ tạo ba người dùng - một cho “Fred” và “Chris” và một cho CEO. CEO sẽ là người dùng chính ở đây và phải có quyền truy cập vào tất cả các dòng dữ liệu trong table. Table Sales tạo phía dưới để demo RLS.

|  |
| --- |
|  |
| CREATE TABLE Sales(      UserName VARCHAR(50),      Country VARCHAR(50),      Sales INT  )    INSERT INTO Sales VALUES ('Fred','USA',10000)  INSERT INTO Sales VALUES ('Chris','USA',9500)  INSERT INTO Sales VALUES ('Tom','France',9600)  INSERT INTO Sales VALUES ('Fred','Spain',9200)  INSERT INTO Sales VALUES ('Chris','Germany',9000) | |



## **Bước 1: Tạo users và cấp quyền**

Ta tạo ba users như đã đề cập trước đó. Ta có thể thực thi tập lệnh sau để tạo người dùng như đã đề cập. Ngoài ra, khi người dùng được tạo, ta cần cấp các quyền đã chọn cho tất cả người dùng một cách rõ ràng. Nếu không, người dùng sẽ không thể truy vấn bất kỳ dữ liệu nào từ bảng

|  |
| --- |
|  |
| CREATE USER CEO WITHOUT LOGIN;  GO  CREATE USER Fred WITHOUT LOGIN;  GO  CREATE USER Chris WITHOUT LOGIN;  GO    GRANT SELECT ON dbo.Sales TO CEO;  GRANT SELECT ON dbo.Sales TO Fred;  GRANT SELECT ON dbo.Sales TO Chris; | |

## **Bước 2: Tạo the inline table-valued function**

Hàm này sẽ kiểm tra user đã đăng nhập vào, và sẽ trả về kết quả theo quyền hạn của login đó.

|  |
| --- |
| CREATE FUNCTION dbo.fn\_SalesSecurity(@UserName AS sysname)      RETURNS TABLE  WITH SCHEMABINDING  AS      RETURN SELECT 1 AS fn\_SalesSecurity\_Result      -- Logic for filter predicate      WHERE @UserName = USER\_NAME()      OR USER\_NAME() = 'CEO';  GO |

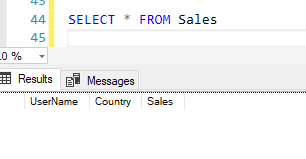
## **Bước 3: Áp dụng Security Policy**

Bước cuối cùng để cài đặt RLS trong SQL Server là áp dụng security policy cụ thể sẽ thực thi FILTER PREDICATE và chuyển nó đến truy vấn trong hàm giống như bộ lọc mệnh đề where

|  |
| --- |
|  |
| CREATE SECURITY POLICY UserFilter  ADD FILTER PREDICATE dbo.fn\_SalesSecurity(UserName)  ON dbo.Sales  WITH (STATE = ON);  GO | |

## **Kiểm tra cơ chế RLS đã cài đặt**

Ta thực hiện lệnh Select truy vấn trên table Sales .



Ta thấy trong hình trên, mặc dù ta đã chèn năm bản ghi vào bảng, nhưng không có bản ghi nào trong số đó hiện đang hiển thị. Điều này là do người dùng mà ta đã đăng nhập vào SQL Server không được xác định trong filter predicate của hàm.

Bây giờ, ta sẽ thực hiện cùng một câu lệnh bằng cách chạy nó trong ngữ cảnh của người dùng “Fred”. Để thay đổi ngữ cảnh người dùng, ta chạy tập lệnh sau.

|  |
| --- |
|  |
| EXECUTE AS USER = 'Fred';  SELECT \* FROM Sales;  REVERT;  GO | |

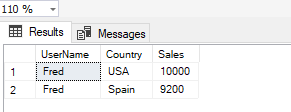
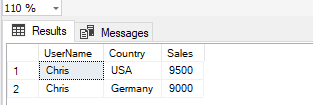


Figure 4 – Executing As Fred

Các bản ghi trả về chỉ cho user “**Fred**”, mặc dù câu lệnh của ta không có mệnh đề Where .

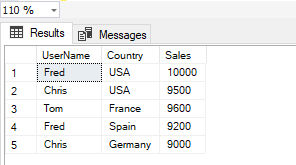
Tương tự, nếu chúng ta thực thi với vai trò user – “**Chris**”:

|  |
| --- |
|  |
| EXECUTE AS USER = 'Chris';  SELECT \* FROM Sales;  REVERT;  GO | |



Nếu chúng ta thực hiện đoạn code trên với user =’CEO’, ta sẽ nhận được tất cả các mẫu tin:

|  |
| --- |
|  |
| EXECUTE AS USER = 'CEO';  SELECT \* FROM Sales;  REVERT;  GO | |



CLS cho phép đảm bảo rằng chỉ những người dùng cụ thể mới có thể xem nội dung của các cột cụ thể trong bảng dữ liệu. CLS có một số thuận lợi quan trọng:

* Kiểm soát truy cập chi tiết — thay vì hạn chế toàn bộ bảng, ta có thể hạn chế quyền truy cập vào cột dữ liệu cụ thể trong đó.
* Có thể dễ dàng thích ứng với các thay đổi đối với các bảng cơ sở dữ liệu, trong khi các điều khiển áp đặt ở cấp ứng dụng có thể bị phá vỡ khi cấu trúc của table thay đổi.
* Không cần các khung nhìn để lọc ra các cột cho người dùng trái phép.

### Để bảo vệ cơ sở dữ liệu ở mức cột, ta dùng lệnh Grant / Deny cấp quyền truy cập cột.

Tùy chọn này cung cấp khả năng kiểm soát chi tiết đối với tính bảo mật của dữ liệu. Không cần thực hiện câu lệnh DENY hoặc GRANT riêng biệt cho mỗi cột. Thay vào đó, ta có thể đưa tên các cột vào lệnh Grant/ Deny. Ví dụ:

GRANT SELECT ON NHANVIEN (MANV, HO, TEN) TO USER1 ;

GO

DENY SELECT ON NHANVIEN (MANV, HO, TEN) TO USER2 ;

#### Auditing

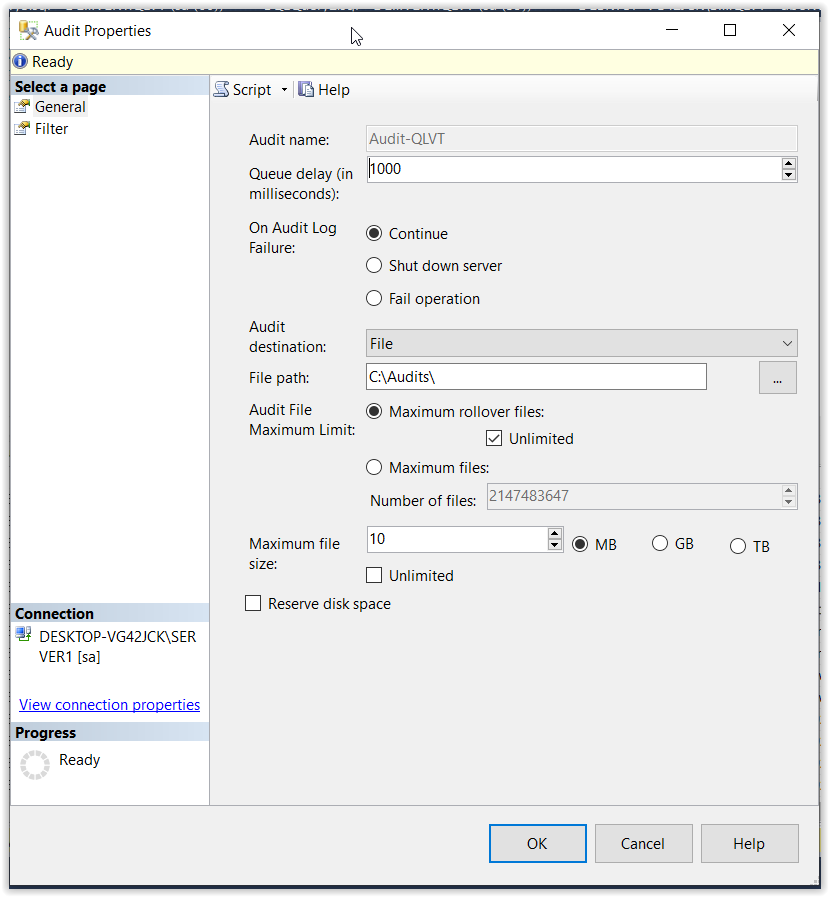
Microsoft cung cấp SQL Server Audit, một công cụ được tích hợp trong SQL Server; công cụ này đọc nhật ký giao dịch và ghi lại thông tin về các thay đổi dữ liệu và đối tượng trong cơ sở dữ liệu vào 1 file. Ta có thể sử dụng nó để truy vấn các truy xuất cấp Server và cấp cơ sở dữ liệu, với các yêu cầu cụ thể.

SQL Server Audit bao gồm các thành phần: SQL Server Audit, SQL Server Audit Specification, Database Audit Specification, và nơi lưu (Target).

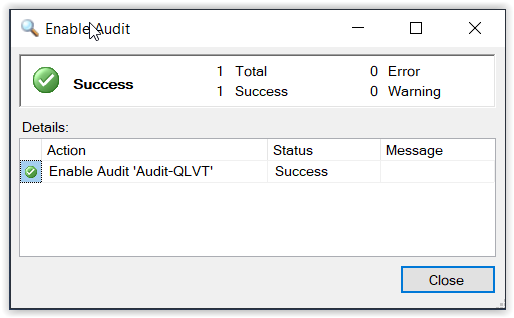
#### SQL Server Audit

SQL Server Audit là một đối tượng thu thập một hành động hoặc một nhóm hành động được yêu cầu giám sát. Quá trình này có thể hoạt động đối với các hành động mức cơ sở dữ liệu hoặc mức server. Ta có thể tạo nhiều kiểm tra cho mỗi Server SQL.

Để tạo 1 audit, ta cần chỉ định nơi chứa các thao tác để về sau có thể truy vấn theo dõi nhật ký các lệnh đã sử dụng trên cơ sở dữ liệu, về cơ bản là vị trí cho đầu ra của kết quả. Audit khi vừa tạo sẽ ở trạng thái chưa hoạt động.



Right click trên Audit vừa tạo, chọn Enable Audit:



#### SQL Server Audit Specification

Đối tượng Server Audit Specification hoạt động ở mức Server. Ta chỉ có thể tạo một đối tượng cho mỗi audit. Đối tượng Server Audit Specification thu thập các event ở mức server.

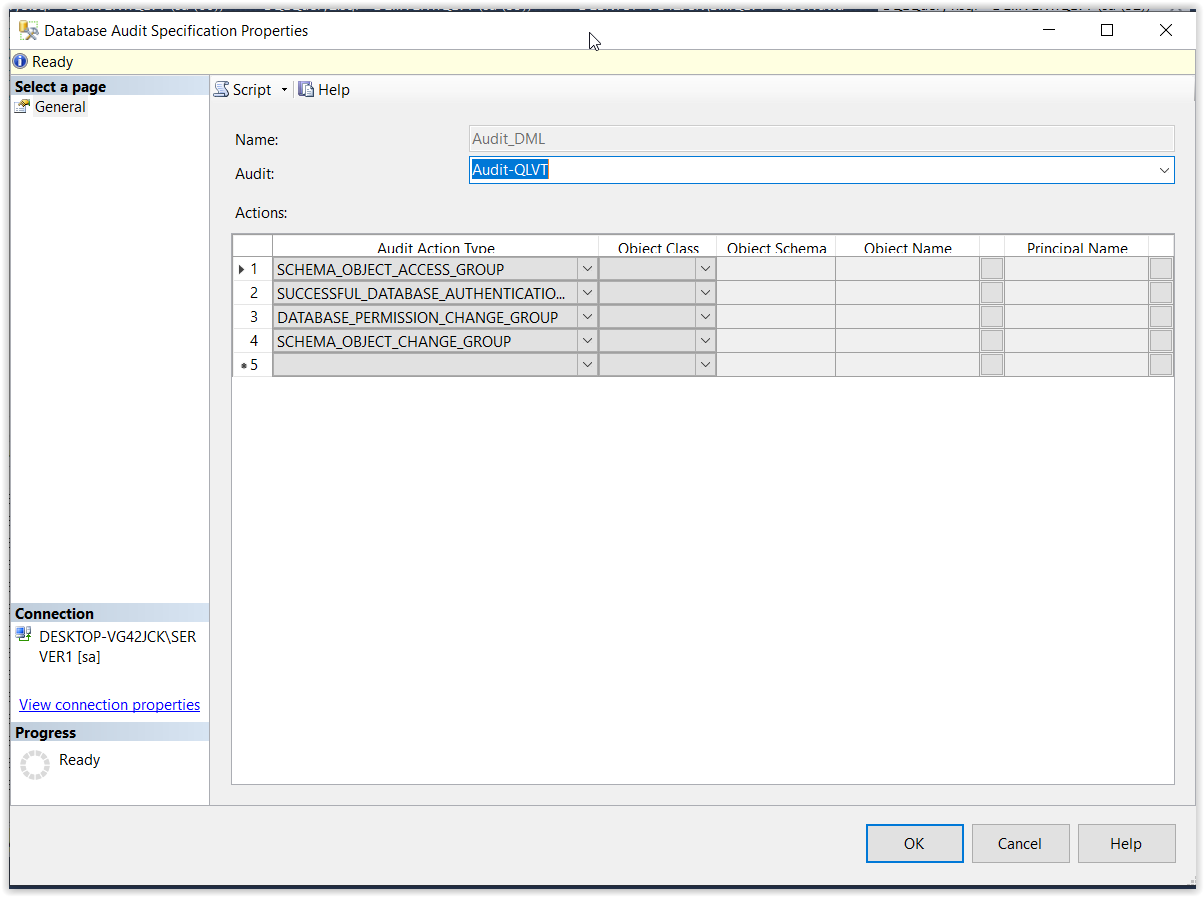
#### Database Audit Specification

Đối tượng Database Audit Specification thu thập các hành động xảy ra mức cơ sở dữ liệu. Ta có thể tạo hai loại audit chính:

• **Audit events** — các hành động được ghi bởi SQL Server engine

• **Audit action groups** —nhóm các hành động được xác định trước

Cả 2 loại audit trên đều được đặt ở mức cơ sở dữ liệu. Ta có thể tạo một database audit cho mỗi cơ sở dữ liệu. Khi ta gửi các hành động đến audit, nó sẽ ghi chúng lại ở file mà ta đã định nghĩa.



#### Target

Kết quả đánh giá được gửi đến một target do ta đã định nghĩa. Ta có thể định nghĩa 1 file như 1 target.

Để xem kết quả đã ghi lại, ta dùng Hàm fn\_get\_audit\_file:

SELECT \* FROM SYS.fn\_get\_audit\_file

('C:\AUDITS\\*.sqlaudit' , default, default

)

#### Mã hóa (Encryption)

* Thông tin truyền được mã hóa: các kết nối tới SQL Server có thể được bảo vệ bằng SSL / TLS. SQL Server truyền dữ liệu bằng giao thức Tabular Data Stream (TDS), giao thức này không được mã hóa theo mặc định. Quản trị viên có thể cung cấp chứng chỉ và bật SSL / TLS, bằng cách bật “Encryption Enabled” trong trang thuộc tính cơ sở dữ liệu hoặc bằng cách chỉ định tùy chọn “Encrypt” trong một chuỗi kết nối ứng dụng cụ thể.

Ví dụ:

string connectionString = "Data Source= DESKTOP-VG42JCK; Initial Catalog=QLVT; Integrated Security=true; **Column Encryption Setting=enabled**";

SqlConnection connection = new SqlConnection(connectionString);

* **Windows Data Protection API (DPAPI)** —đây là khả năng mã hóa được tích hợp sẵn trong Windows — thuật toán mã hóa được sử dụng phụ thuộc vào phiên bản DPAPI có trong Windows. Ở mức SQL Server, các đối tượng bảo mật mà nó có thể sử dụng là tài khoản đăng nhập, nhóm server và thông tin xác thực. Ở mức cơ sở dữ liệu, các đối tượng bảo mật là người dùng.

**Phần 3: Kết luận, đề xuất**

* Kết luận: Chuyên đề đã Tìm hiểu các khả năng dữ liệu không an toàn khi có nhiều người dùng sử dụng chung 1 cơ sở dữ liệu, và đưa ra các phương pháp đảm bảo an toàn dữ liệu trong SQl Server
* Đề xuất (khuyến nghị) : hướng vận dụng kết quả của báo cáo chuyên đề: kết quả nghiên cứu có thể áp dụng trong việc đào tạo, giới thiệu cho sinh viên biết trong các môn học có liên quan đến cơ sở dữ liệu, và Hệ quản trị cơ sở dữ liệu SQL Server.