Quản lý dữ liệu bằng cách sử dụng Microsoft SQL Server

Phiên: 12

Gây nên

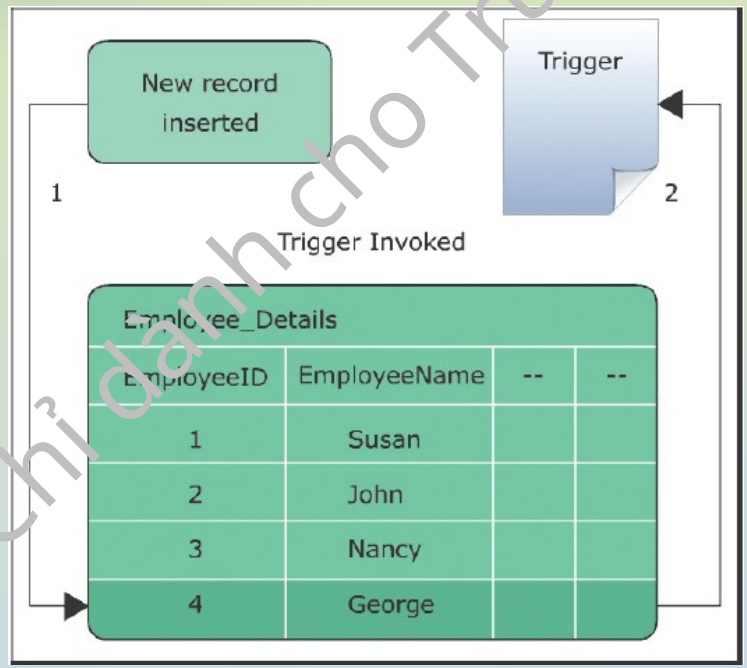
Objectives

- Giải thích các yếu tố kích hoạt
- Giải thích các loại trình kích hoạt khác nhau
- Giải thích quy trình tạo trình kích hoạt DML
- Giải thích quy trình để thay đổi trình kích hoạt DML
- Mô tả các trình kích hoạt lồng nhau
- Mô tả các chức năng cập nhật
- Giải thích việc xử lý nhiều hàng trong một phiên
- Giải thích ý nghĩa hiệu suất của các trình kích hoạt

Introduction

¾Trình kích hoạt:

- là một thủ tục được lưu trữ được thực thi khi cố gắng sửa đổi dữ liệu trong bảng được bảo vệ bởi trình kích hoạt.
- không thể được thực thi trực tiếp, cũng như không truyền hoặc nhận các tham số.
- được định nghĩa trên các bảng cụ thể và các bảng này được gọi là bảng kích hoạt.
- được định nghĩa trênCHÈN, CẬP NHẬT,hoặcXÓA BỎhành động trên bảng, nó sẽ tự động kích hoạt khi những hành động này được thực hiện.
- được tạo bằng cách sử dụngTẠO TRIGGERtuyên bố.



Uses of Triggers

Bộ kích hoạt có thể chứa logic xử lý phức tạp và thường được sử dụng để duy trì tính toàn vẹn của dữ liệu mức thấp. Các mục đích sử dụng chính của trình kích hoạt có thể được phân loại như sau:

Xếp tầng các thay đổi thông qua các bảng liên quan

- Người dùng có thể sử dụng một trình kích hoạt để phân tầng các thay đổi thông qua các bảng liên quan.

Thực thi tính toàn vẹn của dữ liệu phức tạp hơn so với ràng buộc KIỂM TRA

- Không giốngKIỂM TRAcác ràng buộc, trình kích hoạt có thể tham chiếu đến các cột trong các bảng khác.
- Có thể được sử dụng để áp dụng các kiểm tra tính toàn vẹn của dữ liệu phức tạp bằng cách,
  - Kiểm tra các ràng buộc trước khi cập nhật hoặc xóa xếp tầng
  - Tạo trình kích hoạt nhiều hàng cho các hành động được thực thi trên nhiều hàng
  - Thực thi tính toàn vẹn tham chiếu giữa các cơ sở dữ liệu

Xác định thông báo lỗi tùy chỉnh

- Được sử dụng để cung cấp các giải thích phù hợp hoặc chi tiết hơn trong các tình huống lỗi nhất định.

Transact-SQL Programming Elements

Các phần tử lập trình Transact-SQL cho phép thực hiện các hoạt động khác nhau mà không thể thực hiện được trong một câu lệnh duy nhất.

Duy trì dữ liệu không chuẩn hóa

- Tính toàn vẹn của dữ liệu mức thấp có thể được duy trì trong môi trường cơ sở dữ liệu không chuẩn hóa bằng cách sử dụng các trình kích hoạt.
- Dữ liệu không chuẩn hóa thường đề cập đến dữ liệu thừa hoặc dữ liệu có nguồn gốc. Ở đây, trình kích hoạt được sử dụng để kiểm tra không yêu cầu kết quả khớp chính xác.

So sánh trạng thái trước và sau của dữ liệu đang được sửa đổi

- Trình kích hoạt cung cấp tùy chọn để tham chiếu các thay đổi được thực hiện đối với dữ liệu bởiCHÈN, CẬP NHẬT,vàXÓA BỎcác câu lệnh.

Types of Triggers

Trình kích hoạt có thể được đặt để tự động thực hiện một hành động khi một sự kiện ngôn ngữ xảy ra trong một bảng hoặc một dạng xem. Kích hoạt trong SQL Server 2012 có thể được phân loại thành ba loại cơ bản:

Trình kích hoạt DML

- Thực thi khi dữ liệu được chèn, sửa đổi hoặc xóa trong bảng hoặc dạng xem bằng cách sử dụngCHÈN, CẬP NHẬT,hoặcXÓA BỎcác câu lệnh.

Kích hoạt DDL

- Thực thi khi một bảng hoặc dạng xem được tạo, sửa đổi hoặc xóa bằng cách sử dụngCREATE, ALTER hoặcLÀM RƠIcác câu lệnh.

Kích hoạt đăng nhập

- Thực thi các thủ tục được lưu trữ khi một phiên được thiết lập với ĐĂNG NHẬPbiến cố.

DDL Triggers versus DML Triggers

Các trình kích hoạt DDL và DML có các cách sử dụng khác nhau và được thực thi với các sự kiện cơ sở dữ liệu khác nhau.

¾Bảng sau liệt kê một số từ khóa ngôn ngữ điều khiển luồng Transact-SQL:

DDL Triggers	DML Triggers
DDL triggers execute stored procedures on CREATE, ALTER, and DROP statements.	DML triggers execute on INSERT, UPDATE, and DELETE statements.
DDL triggers are used to check and control database operations.	DML triggers are used to enforce business rules when data is modified in tables or views.
DDL triggers operate only after the table or a view is modified.	DML triggers execute either while modifying the data or after the data is modified.
DDL triggers are defined at either the database or the server level.	DML triggers are defined at the database level.

Creating DML Triggers

Trình kích hoạt DML được thực thi khi các sự kiện DML xảy ra trong bảng hoặc dạng xem. Các sự kiện DML này bao gồmCHÈN, CẬP NHẬT,vàXÓA BỎcác câu lệnh.

Trình kích hoạt DML thực thi tính toàn vẹn tham chiếu bằng cách xếp tầng các thay đổi đối với các bảng có liên quan khi một hàng được sửa đổi.

Trình kích hoạt DML có ba loại chính, cụ thể là,CHÈNcò súng,CẬP NHẬTkích hoạt, và XÓA BỎcò súng.

Introduction to Inserted and Deleted Tables

Câu lệnh SQL trong trình kích hoạt DML sử dụng hai loại bảng đặc biệt để sửa đổi dữ liệu trong cơ sở dữ liệu. Các bảng này như sau:

Bảng đã chèn

- Chứa các bản sao của hồ sơ được sửa đổi vớiCHÈN vàCẬP NHẬTcác thao tác trên bảng kích hoạt.
- CácCHÈNvàCẬP NHẬTthao tác chèn các bản ghi mới vào bảng Đã chèn và Kích hoạt.

Bảng đã xóa

- Chứa các bản sao của hồ sơ được sửa đổi vớiXÓA BỎ vàCẬP NHẬTcác thao tác trên bảng kích hoạt.
- Các thao tác này xóa các bản ghi khỏi bảng kích hoạt và chèn chúng vào bảng Đã xóa.

Các bảng Đã chèn và Đã xóa không tồn tại về mặt vật lý trong cơ sở dữ liệu và được tạo và loại bỏ bất cứ khi nào xảy ra bất kỳ sự kiện kích hoạt nào.

Insert Triggers 1-4

Được thực thi khi một bản ghi mới được chèn vào một bảng.

Lưu bản sao của bản ghi đó trong bảng Đã chèn và kiểm tra xem giá trị mới trong bảng Đã chèn có tuân theo các ràng buộc đã chỉ định hay không.

Được tạo bằng cách sử dụngCHÈNtừ khóa trongTẠO TRIGGERvàALTER TRIGGERcác câu lệnh.

Insert Triggers 2-4

Cú pháp:

```
TẠO TRIGGER [schema_name.] Trigger_name ON
[schema_name.] Table_name [WITH ENCRYPTION] {FOR
INSERT} AS
[NẾU CẬP NHẬT (tên_mạch) ...] [{VÀ | HOẶC}
CẬP NHẬT (tên_mạch) ...] <sql_statements>
```

ở đâu,

schema_name:chỉ định tên của lược đồ mà bảng / trình kích hoạt thuộc về. trigger_name:chỉ định tên của trình kích hoạt. tên_bảng:chỉ định bảng mà trình kích hoạt DML được tạo. CÓ TÍCH LŨY:mã hóa văn bản củaTẠO TRIGGERtuyên bố.

VÌ:chỉ định rằng trình kích hoạt DML thực thi sau khi các hoạt động sửa đổi hoàn tất. CHÈN:chỉ định rằng trình kích hoạt DML này sẽ được gọi bằng các thao tác chèn.

CẬP NHẬT:Trả về một giá trị Boolean cho biết liệu mộtCHÈNhoặcCẬP NHẬTcố gắng đã được thực hiện trên một cột cụ thể.

Insert Triggers 3-4

tên cột dọc:Là tên của cột để kiểm traCẬP NHẬThoạt động. VÀ:Kết hợp hai biểu thức Boolean và trả về TRUE khi cả hai biểu thức đều TRUE. HOẶC:Kết hợp hai biểu thức Boolean và trả về TRUE nếu ít nhất một biểu thức là TRUE. Câu lệnh sql:chỉ định các câu lệnh SQL được thực thi trong trình kích hoạt DML.

¾ Đoạn mã sau đây cho biết cách tạo mộtCHÈNkích hoạt trên một bảng có tên **Giao dịch tài khoản**:

```
TẠO TRIGGER Séc Rút tiền_ Số tiền TRÊN Tài
khoản_ Giao dịch
ĐỂ CHÈN

BẰNG
IF (CHỌN Rút tiền từ đã chèn)> 80000 BEGIN

IN 'Số tiền rút không được vượt quá 80000' GIAO DỊCH
ROLLBACK

CHẤM DỨT
```

Insert Triggers 4-4

¾ Đoạn mã sau sẽ chèn một bản ghi và hiển thị thông báo lỗi khi Số tiền rút vượt quá 80000:

```
CHÈN VÀO TÀI KHOẢN_Giao dịch
(ID giao dịch, ID nhân viên, ID khách hàng, ID giao dịch, ngày giao dịch, Số giao dịch, Gửi tiền,
Rút tiền)
GIÁ TRỊ
(1008, 'E08', 'C08', 'T08', '05 / 02/12 ',' TN08 ', 300000,90000)
```

Đầu ra:

```
Số tiền rút không được vượt quá 80000
```

Update Triggers 1-5

Sao chép bản ghi gốc trong bảng Đã xóa và bản ghi mới vào bảng Đã chèn khi bản ghi được cập nhật.

Sao chép bản ghi từ bảng Đã chèn vào bảng kích hoạt miễn là bản ghi hợp lệ.

Được tạo bằng cách sử dụngCẬP NHẬTtừ khóa trongTẠO TRIGGERvàALTER TRIGGERcác câu lệnh.

Update Triggers 2-5

Cú pháp:

```
TẠO TRIGGER [schema_name.] Trigger_name ON
[schema_name.] Table_name [WITH ENCRYPTION] {FOR
UPDATE} AS
[NẾU CẬP NHẬT (tên_mạch) ...] [{VÀ | HOẶC}
CẬP NHẬT (tên_mạch) ...] <sql_statements>
```

ở đâu,

schema_name:chỉ định tên của lược đồ mà bảng / trình kích hoạt thuộc về. trigger_name:chỉ định tên của trình kích hoạt. tên_bảng:chỉ định bảng mà trình kích hoạt DML được tạo. CÓ TÍCH LŨY:mã hóa văn bản củaTẠO TRIGGERtuyên bố.

VÌ:chỉ định rằng trình kích hoạt DML thực thi sau khi các hoạt động sửa đổi hoàn tất. CHÈN:chỉ định rằng trình kích hoạt DML này sẽ được gọi sau các hoạt động cập nhật.

CẬP NHẬT:Trả về một giá trị Boolean cho biết liệu mộtCHÈNhoặcCẬP NHẬTcố gắng đã được thực hiện trên một cột cụ thể.

Update Triggers 3-5

tên cột dọc:Là tên của cột để kiểm traCẬP NHẬThoạt động.

VÀ:Kết hợp hai biểu thức Boolean và trả về TRUE khi cả hai biểu thức đều TRUE. HOẶC: Kết hợp hai biểu thức Boolean và trả về TRUE nếu ít nhất một biểu thức là TRUE.

Câu lệnh sql:chỉ định các câu lệnh SQL được thực thi trong trình kích hoạt DML.

¾ Đoạn mã sau đây cho biết cách tạo mộtCẬP NHẬTkích hoạt ở cấp bảng trên**Thông tin nhân viên**bàn:

```
TẠO TRIGGER CheckBirthDate ON
EmployeeDetails
ĐỂ CẬP NHẬT
BẰNG
IF (CHỌN Ngày sinh từ được chèn)> getDate () BEGIN

IN 'Ngày sinh không được lớn hơn ngày hôm nay' LĂN

CHẤM DỨT
```

Update Triggers 4-5

¾ Đoạn mã sau cập nhật bản ghi và hiển thị thông báo lỗi khi chỉ định ngày sinh không hợp lệ:

```
CẬP NHẬT Nhân viên Chi tiết
SET Ngày sinh = '06/6/2015
'WHERE EmployeeID =' E06 ')
```

Đầu ra:

```
Ngày sinh không được lớn hơn ngày hôm nay.
```

Tạo kích hoạt cập nhật

- Được tạo ở cấp cột hoặc cấp bảng.
- Trình kích hoạt ở cấp cột thực thi khi cập nhật được thực hiện trong cột được chỉ định.
- Trình kích hoạt ở cấp bảng thực thi khi cập nhật được thực hiện ở bất kỳ đâu trong toàn bộ bảng.
- CẬP NHẬT ()hàm được sử dụng để chỉ định cột khi tạoCẬP NHẬT kích hoạt ở cấp cột.

Update Triggers 5-5

¾ Đoạn mã sau tạo ra mộtCẬP NHẬTkích hoạt ở cấp cột trênMã hiệu công nhâncột củaThông tin nhân viênbàn:

```
TẠO TRIGGER Check_E JobeeID TRÊN
EmployeeDetails
ĐỂ CẬP NHẬT

BẢNG
NẾU CẬP NHẬT (ID nhân viên)
BẮT ĐẦU
IN 'Bạn không thể sửa đổi ID của nhân viên' GIAO DỊCH
ROLLBACK

CHẤM DỨT
```

¾Đoạn mã sau khiến trình kích hoạt cập nhật kích hoạt:

```
CẬP NHẬT Nhân viên Chi tiết
SET EmployeeID = 'E12'
WHERE EmployeeID = 'E04'
```

Delete Triggers 1-3

Có thể được tạo để hạn chế người dùng xóa một bản ghi cụ thể trong bảng.

- Bản ghi bị xóa khỏi bảng kích hoạt và được chèn vào bảng Đã xóa.
- Nó được kiểm tra các ràng buộc chống lại việc xóa.
- Nếu có một ràng buộc đối với bản ghi để ngăn việc xóa,XÓA BỎtrình kích hoạt hiển thị thông báo lỗi.
- Bản ghi đã xóa được lưu trữ trong bảng Đã xóa được sao chép trở lại bảng kích hoạt.

Delete Triggers 2-3

Cú pháp:

```
TẠO TRIGGER <trigger_name> BẬT
<table_name>
[CÓ TÍCH LŨY]
ĐỂ XÓA
AS <sql_statement>
```

ở đâu,

XÓA BỎ:chỉ định rằng trình kích hoạt DML này sẽ được gọi bằng các thao tác xóa.

Delete Triggers 3-3

¾ Đoạn mã sau đây cho biết cách tạo mộtXÓA BỎkích hoạt trên **Giao dịch tài khoản**bàn:

```
TẠO TRIGGER Các giao dịch kiểm tra TRÊN
Tài khoản_Giao dịch
ĐỂ XÓA

BẰNG
NẾU 'T01' VÀO (CHỌN ID giao dịch TỪ đã xóa) BẮT ĐẦU

IN 'Người dùng không thể xóa các giao dịch.' GIAO DỊCH
ROLLBACK

CHẤM DỨT
```

¾ Đoạn mã sau đây xóa các bản ghi khỏi**Giao dịch tài khoản** bảng trong đó Tiền gửi là 50000 và hiển thị thông báo lỗi:

```
XÓA KHỎI TÀI KHOẢN_Giao dịch TẠI ĐÂU
Tiền gửi = 50000
```

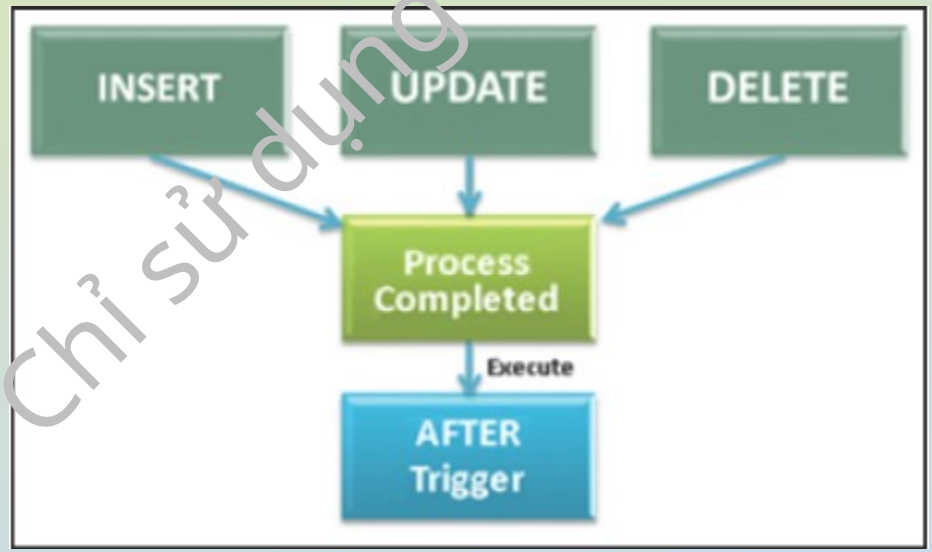
Đầu ra:

```
Người dùng không thể xóa các giao dịch.
```

AFTER Triggers 1-3

Được thực hiện khi hoàn thànhCHÈN, CẬP NHẬT,hoặcXÓA BỎvà chỉ có thể được tạo trên bảng.

Được thực thi khi quá trình kiểm tra ràng buộc trong bảng được hoàn thành và trình kích hoạt cũng được thực thi sau khi bảng Đã chèn và Đã xóa được tạo.



AFTER Triggers 2-3

Cú pháp:

```
TẠO TRIGGER <trigger_name> BẬT
<table_name>
[CÓ TÍCH LŨY]
{CHO | SAU}
{[INSERT] [,] [UPDATE] [,] [DELETE]} NHƯ <sql_statement>
```

ở đâu,

CHO | SAU:chỉ định rằng trình kích hoạt DML thực thi sau khi các hoạt động sửa đổi hoàn tất.

{[CHÈN] [,] [CẬP NHẬT] [,] [XÓA]}:chỉ định các hoạt động gọi trình kích hoạt DML.

AFTER Triggers 3-3

¾ Đoạn mã sau đây cho biết cách tạo mộtSAU KHI XÓAkích hoạt trên **Thông tin nhân viên**bàn:

```
TẠO TRIGGER Employee_Deletion ON
EmployeeDetails
SAU KHI XÓA
BẰNG
BẮT ĐẦU
KHAI BÁO @num nchar;
CHỌN @num = COUNT (*) TỪ IN đã xóa 'Không. nhân
viên bị xóa = '+ @num END
```

¾ Đoạn mã sau sẽ xóa một bản ghi khỏi**Thông tin nhân viên** bảng và hiển thị thông báo lỗi:

```
XÓA khỏi EmployeeDetails WHERE EmployeeID = 'E07'
```

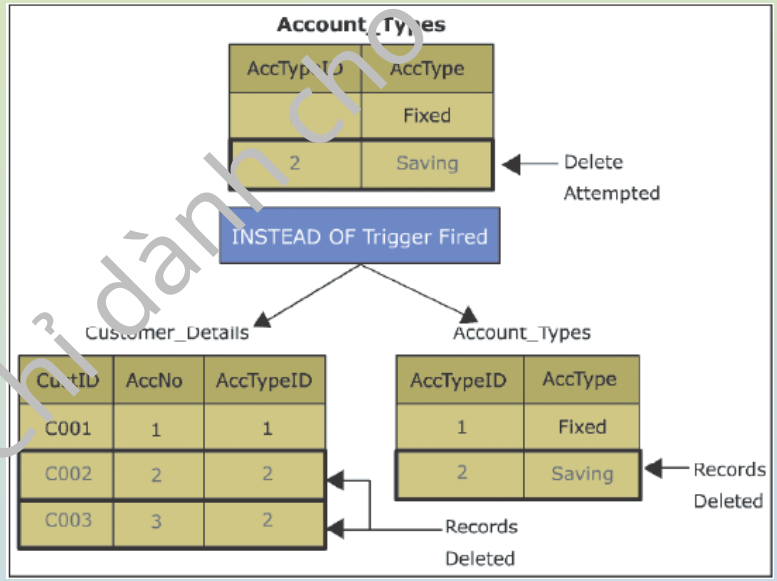
Đầu ra:

```
Không. nhân viên bị xóa = 0.
```

INSTEAD OF Triggers 1-3

Được thực hiện thay choCHÈN, CẬP NHẬT,hoặcXÓA BỎcác hoạt động.

Được thực hiện trước khi kiểm tra ràng buộc được thực hiện trên bảng và sau khi tạo bảng Đã chèn và Đã xóa.



INSTEAD OF Triggers 2-3

Cú pháp:

```
TẠO TRIGGER <trigger_name> BẬT
{<table_name> | <view_name>} {CHO | SAU
| THAY VÌ}
{[INSERT] [,] [UPDATE] [,] [DELETE]} NHƯ <sql_statement>
```

ở đâu,

view_name:chỉ định chế độ xem mà trình kích hoạt DML được tạo.

THAY VÌ:chỉ định rằng trình kích hoạt DML thực thi thay cho các hoạt động sửa đổi. Các trình kích hoạt này không được xác định bằng các chế độ xem có thể cập nhậtVỚI LỰA CHỌN KIỂM TRA.

INSTEAD OF Triggers 3-3

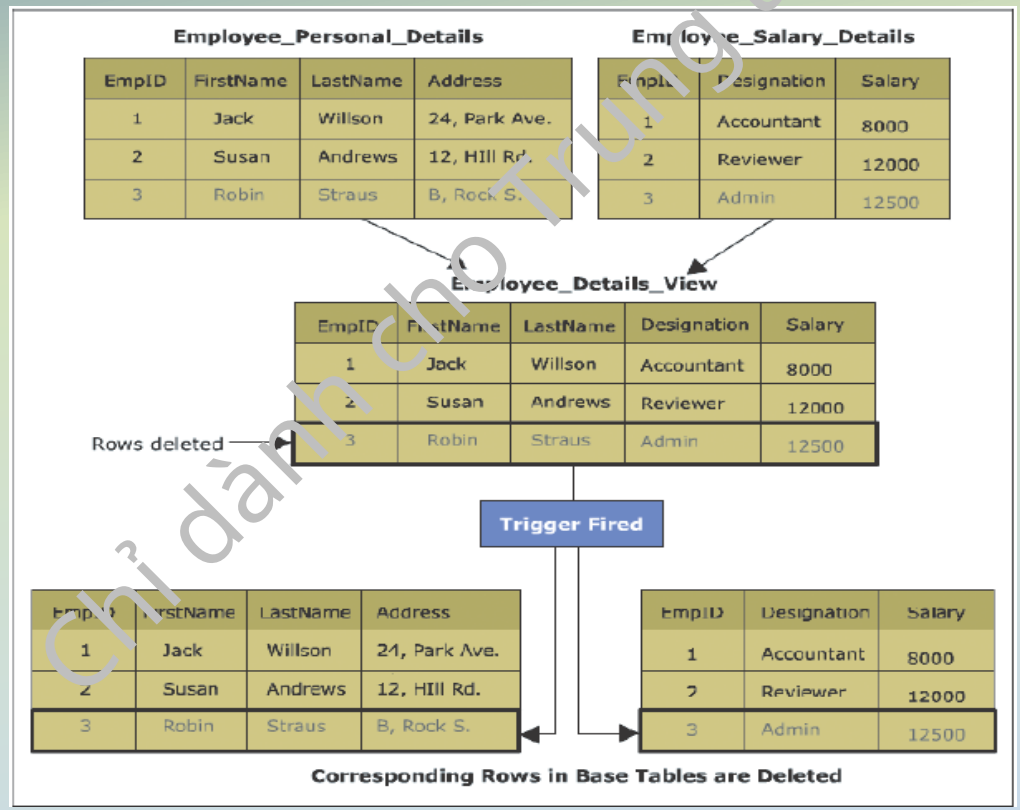
¾ Đoạn mã sau tạo ra mộtINSTEAD OF DELETEkích hoạt trên **Giao dịch tài khoản**bàn:

```
TẠO TRIGGER Xóa_AccType TRÊN Tài
khoản_Giao dịch
INSTEAD OF DELETE
BẰNG
BẮT ĐẦU
XÓA KHỎI NHÂN VIÊN

XÓA KHỎI TÀI KHOẢN_Trans Giao dịch TẠI ĐẦU GIAO DỊCHTypeID VÀO (CHỌN
GIAO DỊCHTypeID TỪ đã bị xóa)
CHẤM DỨT
```

Using INSTEAD OF Triggers with Views 1-3

Có thể được chỉ định trên bảng cũng như dạng xem và cung cấp phạm vi rộng hơn và các loại cập nhật mà người dùng có thể thực hiện đối với dạng xem.



Using INSTEAD OF Triggers with Views 2-3

¾ Đoạn mã sau tạo một bảng có tên**Employee_Personal_Details**:

```
TẠO BẢNG Employee_Personal_Details (

EmpID int NOT NULL, FirstName varchar
(30) NOT NULL, LastName varchar (30)
NOT NULL, Address varchar (30)

)
```

¾ Đoạn mã sau tạo một bảng có tên**Employee_Salary_Details**:

```
TẠO BẢNG Employee_Salary_Details (

EmpID int NOT NULL, Design
varchar (30), Lương int NOT
NULL
)
```

Using INSTEAD OF Triggers with Views 3-3

¾ Đoạn mã sau tạo một chế độ xem từ bảng có tên **Employee_Personal_Details**và**Employee_Salary_Details**:

```
TẠO CHẾ ĐỘ XEM Employee_Details_View
AS
CHỌN e1.EmpID, FirstName, LastName, Chỉ định, Mức lương TỪ
Employee_Personal_Details e1
THAM GIA Employee_Salary_Details e2
ON e1.EmpID = e2.EmpID
```

¾ Đoạn mã sau tạo ra mộtINSTEAD OF DELETEcò súng **Xóa_người lao động**trên quan điểm:

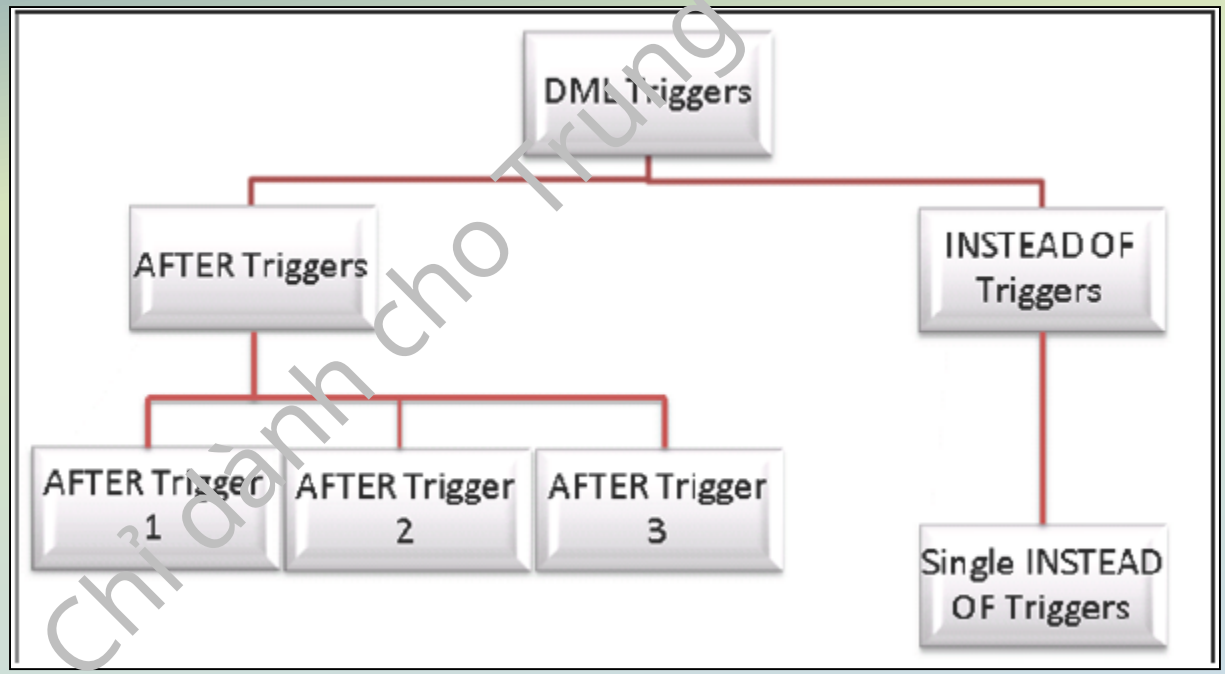
```
TẠO TRIGGER Xóa_người lao động TRÊN
Employee_Details_View
INSTEAD OF DELETE
BẰNG
BẮT ĐẦU
DELETE FROM Employee_Salary_Details WHERE EmpID IN (CHỌN
EmpID FROM đã bị xóa)
DELETE FROM Employee_Personal_Details WHERE EmpID IN (CHỌN
EmpID FROM đã bị xóa)
```

¾Đoạn mã sau sẽ xóa một hàng khỏi chế độ xem:

```
XÓA KHỎI Employee_Details _View WHERE EmpID = '3'
```

Working with DML Triggers 1-3

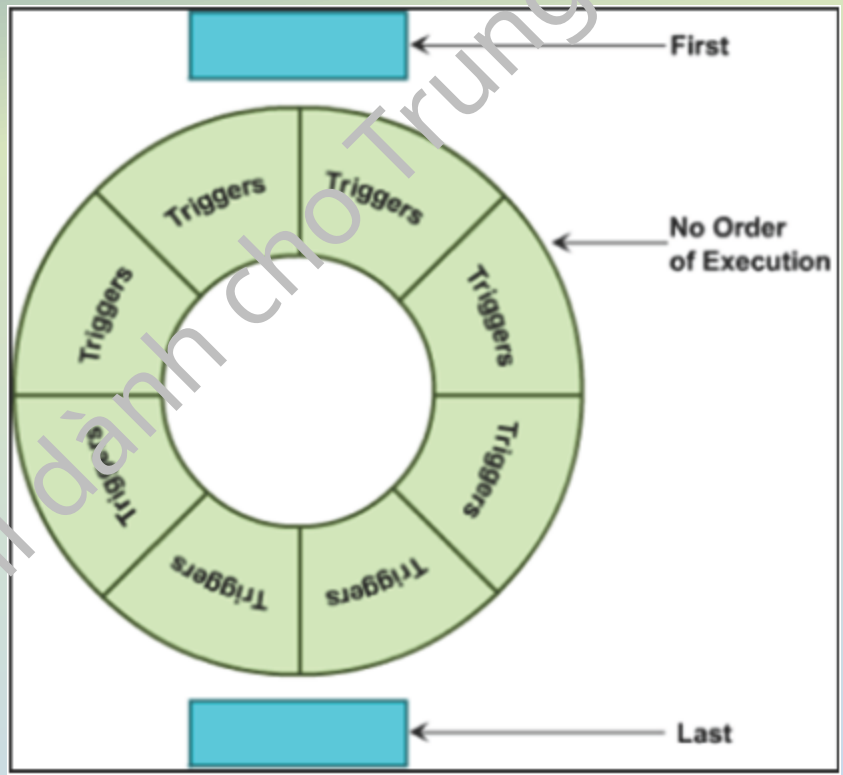
Khi người dùng có nhiềuSAUtrình kích hoạt là một hành động kích hoạt, tất cả các trình kích hoạt này phải có một tên khác.



Working with DML Triggers 2-3

¾Thứ tự thực thi của trình kích hoạt DML

SQL Server 2012 cho phép người dùng chỉ địnhSAUkích hoạt sẽ được thực hiện đầu tiên và sẽ được thực hiện sau cùng.



Working with DML Triggers 3-3

Cú pháp:

```
sp_settriggerorder [@triggername =] '[triggerschema. ] triggername ', [@order =]' value '

, [@stmttype =] 'statement_type'
```

ở đâu,

[triggerchema.] triggername:là tên của trình kích hoạt DML hoặc DDL và lược đồ chứa nó và thứ tự của nó cần được chỉ định.

giá trị:chỉ định thứ tự thực thi của trình kích hoạt làĐẦU TIÊN, CUỐI CÙNG,hoặcKHÔNG AI.Nếu ĐẦU TIÊN được chỉ định, sau đó trình kích hoạt được kích hoạt trước.

statement_type:chỉ định kiểu câu lệnh SQL (CHÈN, CẬP NHẬT,hoặc XÓA BỎ)sẽ gọi trình kích hoạt DML.

¾ Đoạn mã sau thực thi**Employee_Deletion**trình kích hoạt được xác định trên bảng khi**XÓA BỎ**hoạt động được thực hiện:

```
EXEC sp_settriggerorder @triggername = 'Employee_Deletion', @order =
    'FIRST', @stmttype = 'XÓA
```

Viewing Definitions of DML Triggers

Định nghĩa trình kích hoạt bao gồm tên trình kích hoạt, bảng mà trình kích hoạt được tạo, các hành động kích hoạt và các câu lệnh SQL được thực thi.

Tên trình kích hoạt DML phải được chỉ định làm tham số khi thực thi sp_helptext.

Cú pháp:

```
sp_helptext '<DML_trigger_name>'
```

ở đâu,

DML_trigger_name:chỉ định tên của trình kích hoạt DML có các định nghĩa sẽ được hiển thị.

¾Đoạn mã sau tạo một bảng có tên**Employee_Salary_Details**:

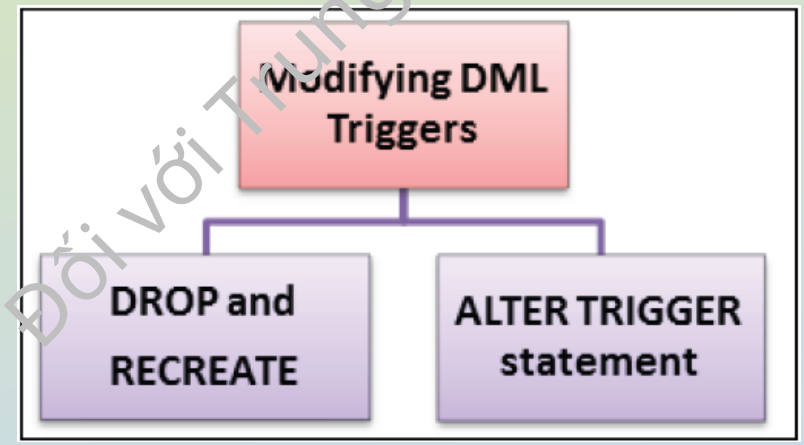
```
sp_helptext 'Employee_Deletion'
```

Modifying Definitions of DML Triggers 1-3

Tham số trình kích hoạt được xác định tại thời điểm tạo trình kích hoạt và bao gồm loại hành động kích hoạt gọi trình kích hoạt và các câu lệnh SQL được thực thi.

- Thả và tạo lại trình kích hoạt với các tham số mới.
- Thay đổi các thông số bằng cách sử dụngALTER TRIGGERtuyên bố.

Trình kích hoạt DML có thể được mã hóa để ẩn định nghĩa của nó.



Modifying Definitions of DML Triggers 2-3

Cú pháp:

```
ALTER TRIGGER <trigger_name> BẬT
{<table_name> | <view_name>} [CÓ TÍCH
LŨY]
{CHO | SAU | THAY VÌ}
{[INSERT] [,] [UPDATE] [,] [DELETE]} NHƯ <sql_statement>
```

ở đâu,

CÓ TÍCH LŨY:chi tiết cụ thể mà các định nghĩa kích hoạt DML không được hiển thị.

CHO | SAU:chỉ định rằng trình kích hoạt DML thực thi sau khi các hoạt động sửa đổi hoàn tất.

THAY VÌ:chỉ định rằng trình kích hoạt DML thực thi thay cho các hoạt động sửa đổi.

Modifying Definitions of DML Triggers 3-3

¾ Đoạn mã sau làm thay đổi**CheckE JobeeID**kích hoạt được tạo trên **Thông tin nhân viên**bảng sử dụngCÓ TÍCH LŨYlựa chọn:

```
ALTER TRIGGER Kiểm tra ID nhân viên
trên Nhân viên
CÓ TÍCH LŨY
ĐỂ CHÈN

BẰNG
NẾU 'E01' VÀO (CHỌN ID Nhân viên TỪ được chèn) BẮT ĐẦU

IN 'Người dùng không thể chèn khách hàng của GIAO DỊCH
ROLLBACK của Áo

CHẤM DỨT
```

Đầu ra:

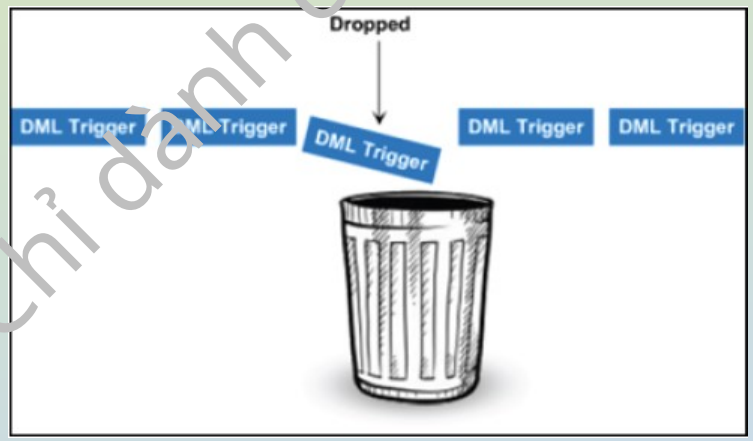
Văn bản cho đối tượng CheckEmployeeID được mã hóa.

Dropping DML Triggers 1-2

Kích hoạt có thể được loại bỏ bằng cách sử dụngDROP TRIGGERtuyên bố.

- Khi một bảng bị loại bỏ, tất cả các trình kích hoạt được xác định trên bảng đó cũng bị loại bỏ.

Khi trình kích hoạt DML bị xóa khỏi bảng, thông tin về trình kích hoạt cũng bị xóa khỏi các dạng xem danh mục.



Dropping DML Triggers 2-2

Cú pháp:

```
DROP TRIGGER <DML_trigger_name> [, ... n]
```

ở đâu,

DML_trigger_name:chỉ định tên của trình kích hoạt DML sẽ được loại bỏ. [, ... N]: các chi tiết cụ thể mà nhiều trình kích hoạt DML có thể bị loại bỏ.

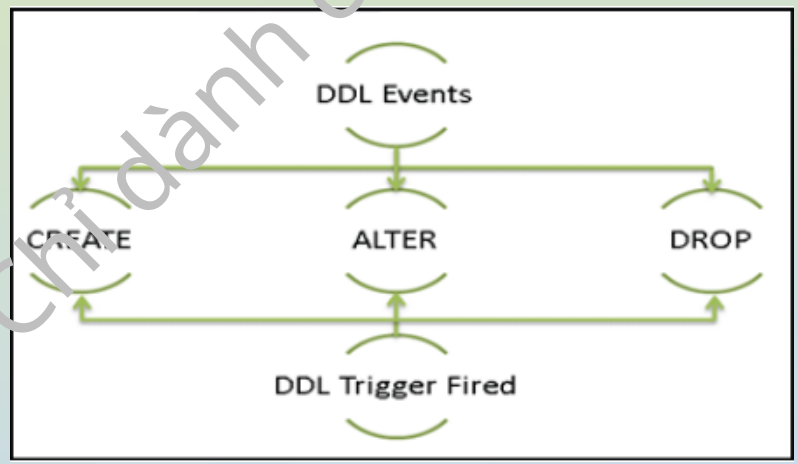
¾ Đoạn mã sau làm giảm**CheckE JoheeD**kích hoạt được tạo trên **Thông tin nhân viên**bàn:

```
DROP TRIGGER Kiểm tra
```

DDL Triggers 1-2

Ngôn ngữ định nghĩa dữ liệu (DDL) kích hoạt thực thi các thủ tục được lưu trữ khi các sự kiện DDL nhưCREATE, ALTER,vàLÀM RƠIcác câu lệnh xảy ra trong cơ sở dữ liệu hoặc máy chủ.

Các trình kích hoạt DDL có thể được sử dụng để ngăn chặn các sửa đổi trong lược đồ cơ sở dữ liệu. Lược đồ là một tập hợp các đối tượng như bảng, dạng xem, v.v. trong cơ sở dữ liệu.



DDL Triggers 2-2

Cú pháp:

```
TẠO TRIGGER <trigger_name> TRÊN
{TẤT CẢ CÁC MÁY CHỦ | CƠ SỞ DỮ
LIỆU} [CÓ TÍCH LŨY]
{CHO | SAU} {<event_type>} NHƯ
<sql_statement>
```

ở đâu,

TẤT CẢ MÁY CHỦ:chỉ định rằng trình kích hoạt DDL thực thi khi các sự kiện DDL xảy ra trong máy chủ hiện tại.

CƠ SỞ DỮ LIỆU:chỉ định rằng trình kích hoạt DDL thực thi khi các sự kiện DDL xảy ra trong cơ sở dữ liệu hiện tại.

loại sự kiện:chỉ định tên của sự kiện DDL gọi kích hoạt DDL.

¾ Đoạn mã sau tạo một trình kích hoạt DDL để giảm và thay đổi bảng:

```
TẠO TRIGGER Bảo mật
TRÊN CƠ SỞ DỮ LIỆU
CHO DROP_TABLE, ALTER_TABLE AS

PRINT 'Bạn phải tắt Trigger "Secure" để thả hoặc thay đổi bảng!' ROLLBACK
```

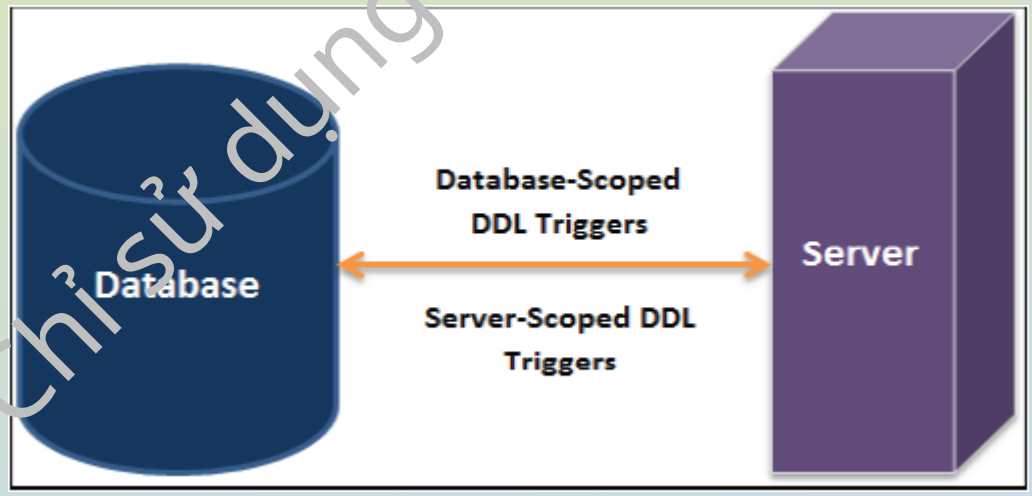
Scope of DDL Triggers 1-2

Các trình kích hoạt DDL được gọi bởi các câu lệnh SQL được thực thi trong cơ sở dữ liệu hiện tại hoặc trên máy chủ hiện tại.

Phạm vi của trình kích hoạt DDL phụ thuộc vào việc trình kích hoạt thực thi cho các sự kiện cơ sở dữ liệu hoặc sự kiện máy chủ.

Scope of DDL Triggers 2-2

- Kích hoạt DDL Phạm vi Cơ sở dữ liệu:
  - được gọi bởi các sự kiện sửa đổi lược đồ cơ sở dữ liệu.
  - lưu trữ các trình kích hoạt trong cơ sở dữ liệu và thực thi trên các sự kiện DDL, ngoại trừ các sự kiện liên quan đến bảng tạm thời.
- Kích hoạt DDL Phạm vi Máy chủ:
  - được gọi bởi các sự kiện DDL ở cấp máy chủ.
  - được lưu trữ trong cơ sở dữ liệu chính.



Nested Triggers 1-2

Cả hai trình kích hoạt DDL và DML được lồng vào nhau khi một trình kích hoạt thực hiện một hành động khởi tạo một trình kích hoạt khác.

Các trình kích hoạt lồng nhau có thể được sử dụng để thực hiện các chức năng như lưu trữ bản sao lưu của các hàng bị ảnh hưởng bởi các hành động trước đó.

Người dùng có thể tắt các trình kích hoạt lồng nhau, bằng cách đặt tùy chọn trình kích hoạt lồng nhau của sp_configurevề 0 hoặc tắt.

Nested Triggers 2-2

¾ Đoạn mã sau tạo ra mộtSAU KHI XÓAkích hoạt có tên **Employee_Deletion**trên**Employee_Personal_Details**bàn:

```
TẠO TRIGGER Employee_Deletion TRÊN
Employee_Personal_Details SAU KHI XÓA

BẰNG
BẮT ĐẦU
PRINT 'Xóa sẽ ảnh hưởng đến bảng Employee_Salary_Details' XÓA TỪ
Employee_Salary_Details WHERE EmpID IN (CHỌN EmpID TỪ đã xóa)

CHẤM DỨT
```

UPDATE 1-2

Trả về giá trị Boolean chỉ định liệu một CẬP NHẬTthoặcCHÈNhành động được thực hiện trên một xem hoặc cột của một bảng.

Có thể được sử dụng ở bất cứ đâu bên trong phần thân của Transact-SQLCẬP NHẬThoặc CHÈNkích hoạt để kiểm tra xem trình kích hoạt có nên thực hiện một số hành động.

Cú pháp:

CẬP NHẬT (cột)

ở đâu,

cột:là tên của cột để kiểm tra mộtCHÈNhoặcCẬP NHẬThoạt động.

UPDATE 2-2

¾ Đoạn mã sau đây tạo ra một trình kích hoạtKế toántrên **Giao dịch tài khoản**bảng để cập nhật các cột**ID giao dịch**hoặc **Mã hiện công nhân**:

```
TẠO TRIGGER Kế toán TRÊN Tài
khoản_Giao dịch SAU KHI CẬP
NHẬT
BẰNG
NẾU (CẬP NHẬT (ID giao dịch) HOẶC CẬP NHẬT (ID nhân viên)) BẮT
ĐẦU
RAISERROR (50009, 16, 10) HẾT;

ĐI
```

Handling of Multiple Rows in a Session

Khi người dùng viết mã cho trình kích hoạt DML, thì câu lệnh khiến trình kích hoạt kích hoạt sẽ là một câu lệnh duy nhất.

Khi chức năng của trình kích hoạt DML liên quan đến việc tự động tính toán lại các giá trị tóm tắt của một bảng và lưu trữ kết quả trong một bảng khác, thì việc xem xét nhiều cụm là rất quan trọng.

¾Đoạn mã sau lưu trữ tổng số đang chạy cho một lần chèn một hàng:

```
SỬ DỤNG AdventureWorks2012;
ĐI
TẠO PODetails TRIGGER
BẬT Mua hàng. Mua hàng Đặt hàng Chi tiết
SAU KHI CHÈN NHƯ
CẬP NHẬT PurchaseOrderHeader
SET SubTotal = SubTotal + LineTotal TỪ được
chèn
WHERE PurchaseOrderHeader.PurchaseOrderID = đã chèn.PurchaseOrderID;
```

Performance Implication of Triggers

Kích hoạt không mang theo chi phí, thay vào đó chúng khá nhạy.

Một quy tắc tốt sẽ là giữ cho logic đơn giản trong các trình kích hoạt và tránh sử dụng con trỏ trong khi thực hiện các câu lệnh chống lại một bảng khác và các tác vụ khác gây ra tốc độ chậm lại.

Summary

- Trình kích hoạt là một thủ tục được lưu trữ được thực thi khi cố gắng sửa đổi dữ liệu trong bảng được trình kích hoạt bảo vệ.
- Trình kích hoạt đăng nhập thực thi các thủ tục được lưu trữ khi một phiên được thiết lập với sự kiện LOGON.
- Trình kích hoạt DML được thực thi khi các sự kiện DML xảy ra trong bảng hoặc dạng xem.
- Kích hoạt CHÈN được thực thi khi một bản ghi mới được chèn vào bảng.
- Trình kích hoạt CẬP NHẬT sao chép bản ghi gốc trong bảng Đã xóa và bản ghi mới vào bảng Đã chèn khi bản ghi được cập nhật.
- Có thể tạo trình kích hoạt XÓA để hạn chế người dùng xóa một bản ghi cụ thể trong bảng.
- Kích hoạt SAU KHI hoàn thành các thao tác CHÈN, CẬP NHẬT hoặc XÓA.