



THỰC HỌC – THỰC NGHIỆP

QUẢN TRỊ CƠ SỞ DỮ LIỆU VỚI SQL SERVER

BÀI 6: TRIGGER

- ⦿ Tổng quan Trigger
- ⦿ Làm việc với trigger



Tổng quan trigger

- ❖ Giới thiệu
- ❖ Các loại trigger
- ❖ Các trigger DML

Làm việc với trigger

- ❖ Trigger AFTER
- ❖ Trigger INSTEAD OF

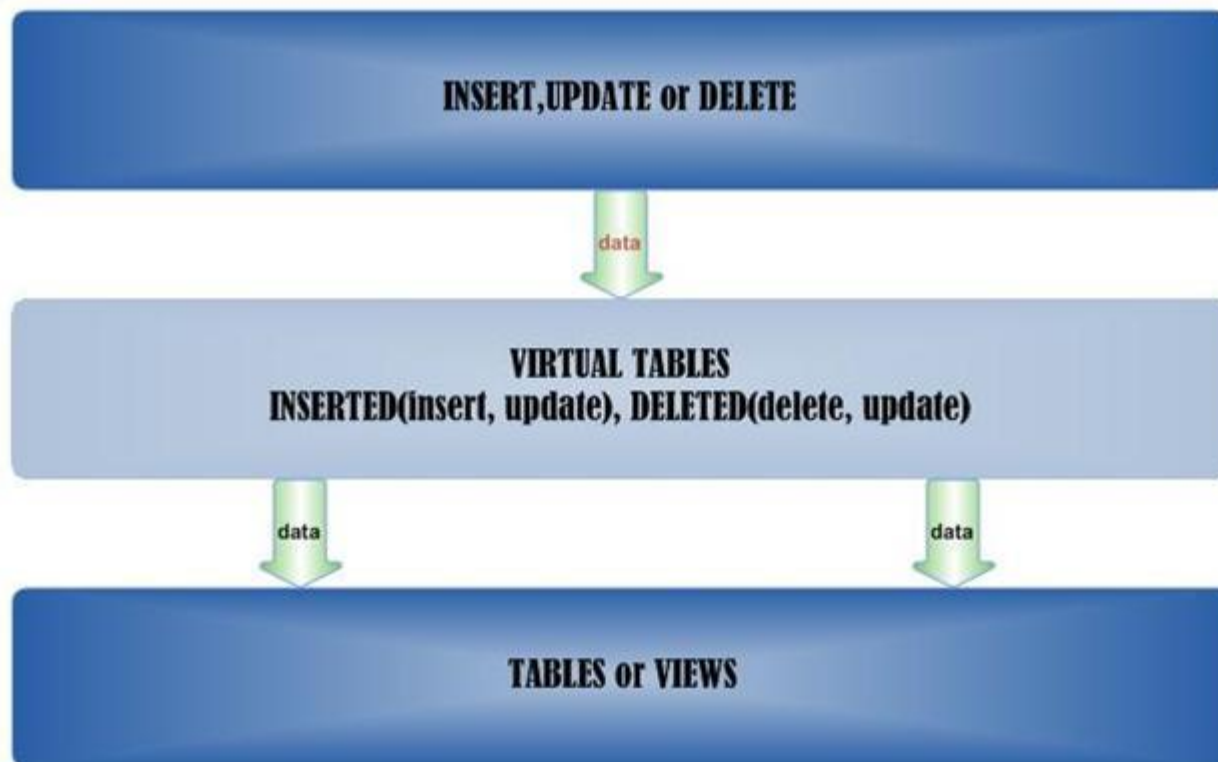




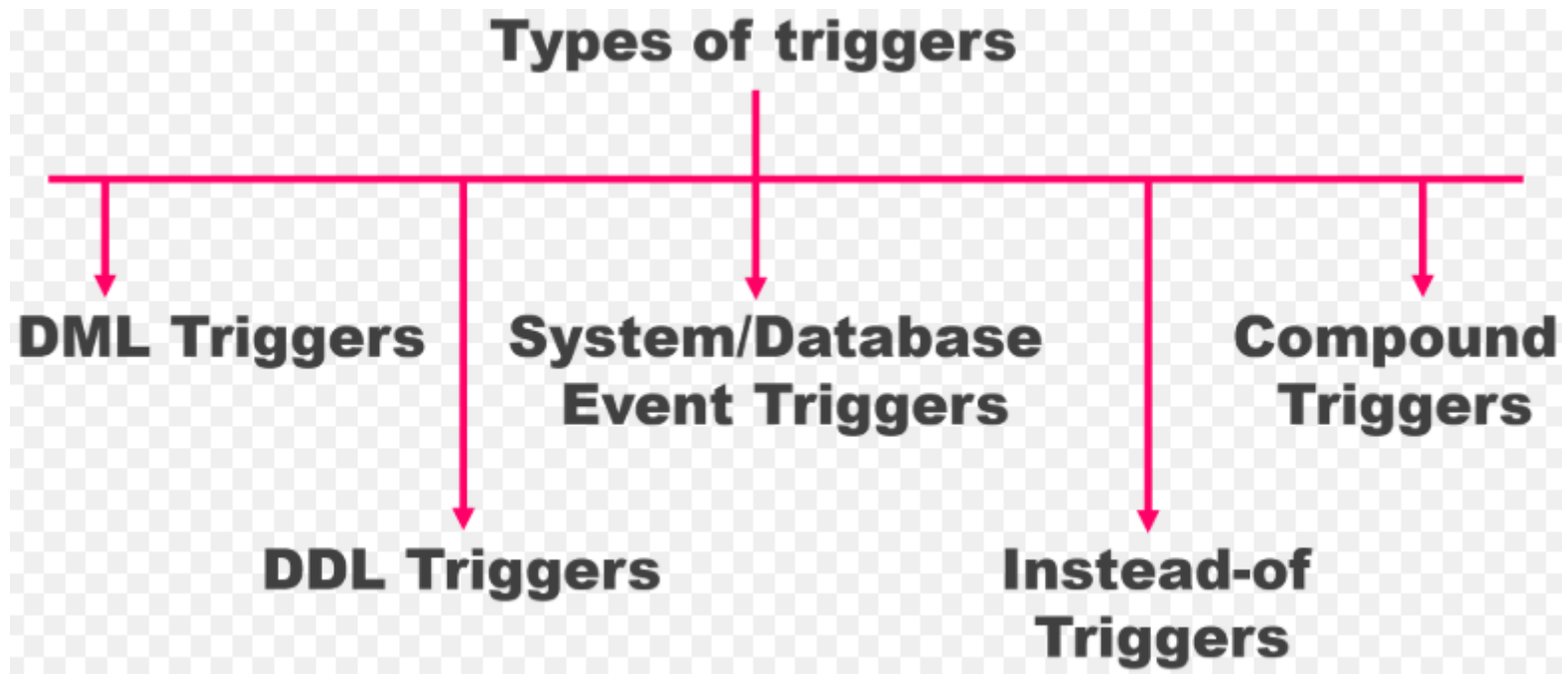
PHẦN 1

- ❑ Trigger là một dạng đặc biệt của thủ tục lưu trữ (store procedure), được thực thi một cách tự động khi có sự thay đổi dữ liệu (do tác động của câu lệnh INSERT, UPDATE, DELETE) trên một bảng nào đó.
- ❑ Không thể gọi thực hiện trực tiếp Trigger bằng lệnh EXECUTE.
- ❑ Trigger là một stored procedure không có tham số.
- ❑ Trigger được lưu trữ trong DB Server và thường hay được dùng để kiểm tra ràng buộc toàn vẹn dữ liệu.

- Thông thường dữ liệu các thao tác insert, update, delete được sao lưu vào bảng tạm inserted và deleted



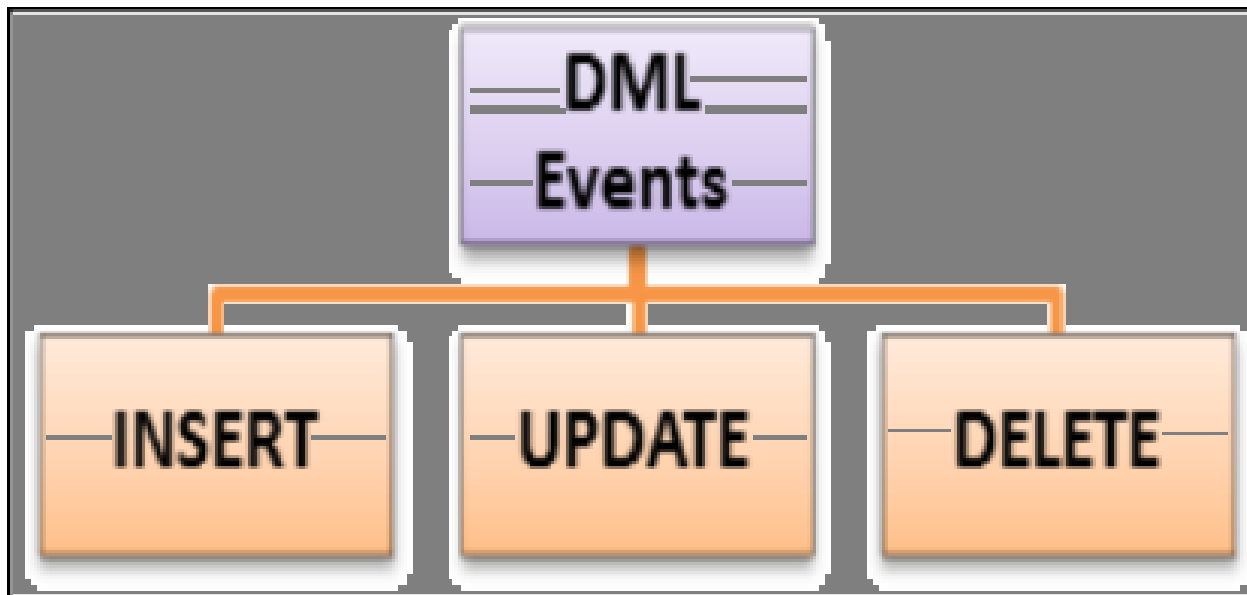
□ Các loại trigger thường dùng



- ❑ So sánh trigger DDL và trigger DML: Các trigger DDL và DML có cách sử dụng khác nhau và được thực thi với các sự kiện cơ sở dữ liệu khác nhau.

Trigger DDL	Trigger DML
Các trigger DDL thực thi các thủ tục lưu trữ trên câu lệnh CREATE, ALTER và DROP.	Các trigger DML thực thi trên các câu lệnh INSERT, UPDATE và DELETE.
Các trigger DDL được sử dụng để kiểm tra và kiểm soát các hoạt động của cơ sở dữ liệu.	Các trigger DML được sử dụng để thực thi các quy tắc thương mại khi dữ liệu được sửa đổi trong các bảng hoặc khung nhìn.
Các trigger DDL chỉ hoạt động sau khi bảng hoặc khung nhìn được sửa đổi.	Các trigger DML thực thi trong hoặc sau khi dữ liệu được sửa đổi.
Các trigger DDL được định nghĩa ở mức cơ sở dữ liệu hoặc máy chủ.	Các trigger DML được định nghĩa ở mức cơ sở dữ liệu.

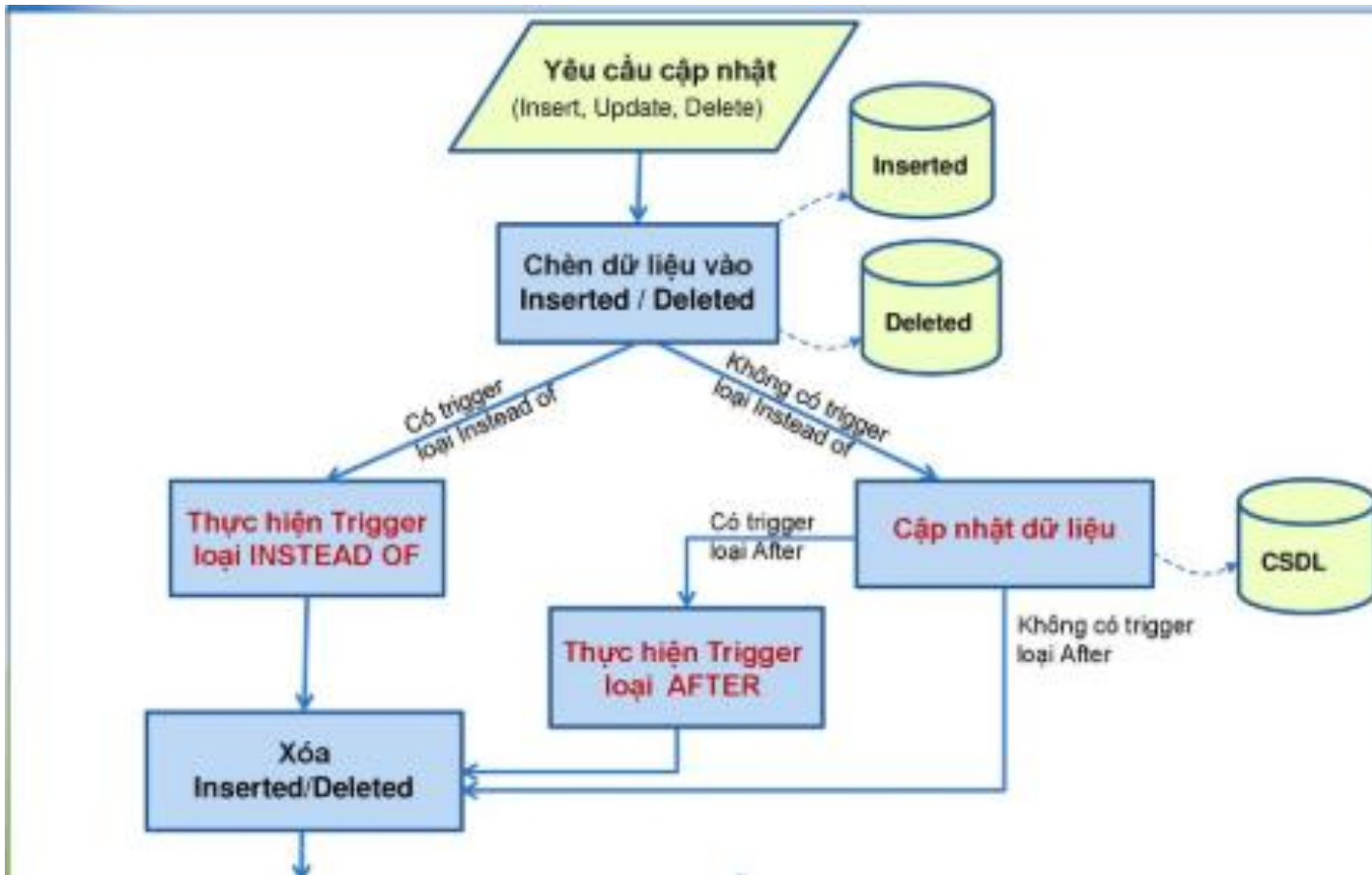
- ❑ Các trigger DML được thực thi khi sự kiện DML xảy ra trong các **bảng hoặc khung nhìn**.
- ❑ Trigger DML này bao gồm các câu lệnh INSERT, UPDATE và DELETE.
- ❑ Các trigger DML gồm ba loại chính: Trigger INSERT, Trigger UPDATE, Trigger DELETE



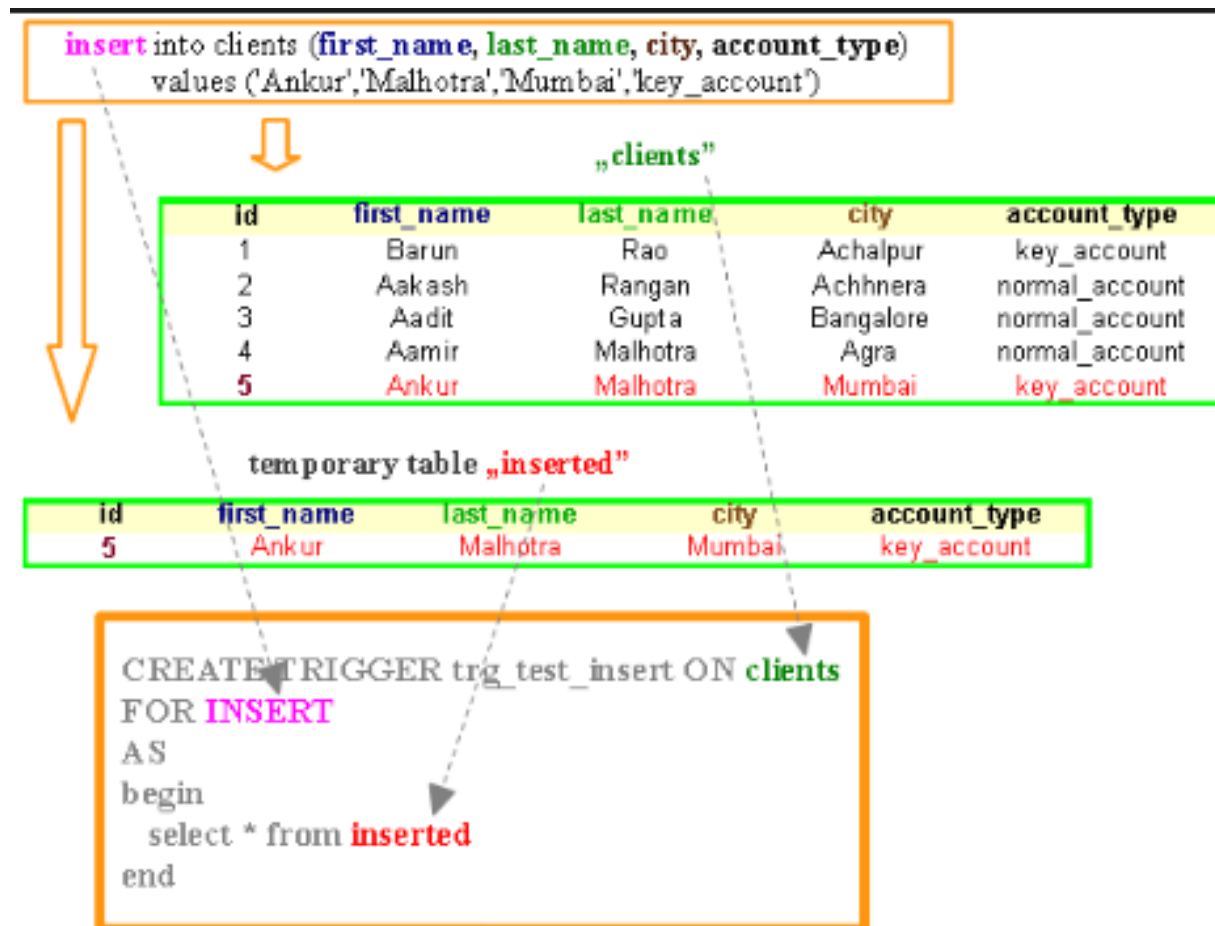
❑ Các bảng Inserted và Deleted

- ❖ Các trigger DML sử dụng hai loại bảng đặc biệt để sửa đổi dữ liệu trong cơ sở dữ liệu.
- ❖ Các bảng tạm thời lưu trữ dữ liệu ban đầu cũng như dữ liệu đã sửa đổi. Những bảng này gồm Inserted và Deleted.
- ❖ Bảng Inserted: chứa bản sao các bản ghi được sửa đổi với hoạt động INSERT và UPDATE trên bảng trigger. Hoạt động INSERT và UPDATE sẽ tiến hành chèn các bản ghi mới vào bảng Inserted và bảng trigger.
- ❖ Bảng Deleted: chứa bản sao của các bản ghi được sửa đổi với hoạt động DELETE và UPDATE trên bảng trigger.

□ Các bảng Inserted và Deleted



□ Các bảng Inserted và Deleted



❑ Trigger INSERT

- ❖ Trigger INSERT được thực thi khi một bản ghi mới được chèn vào bảng
- ❖ Trigger INSERT đảm bảo rằng giá trị đang được nhập phù hợp với các ràng buộc được định nghĩa trên bảng đó.
- ❖ Bảng Inserted và Deleted về khía cạnh vật lý chúng không tồn tại trong cơ sở dữ liệu
- ❖ Trigger INSERT được tạo ra bằng cách sử dụng từ khóa INSERT trong câu lệnh CREATE TRIGGER và ALTER TRIGGER.

□ Trigger INSERT

```
CREATE TRIGGER tên_trigger ON tên_bảng  
FOR {DELETE, INSERT, UPDATE}  
AS  
    câu_lệnh_sql
```

- ❖ tên_trigger: chỉ ra tên của trigger do người dùng tự đặt
- ❖ Tên bảng: chỉ ra bảng mà trên đó trigger DML được tạo ra (bảng trigger).
- ❖ FOR : hoạt động thao tác dữ liệu.
- ❖ Câu lệnh sql: chỉ ra các câu lệnh SQL được thực thi trong trigger DML

- ❑ Ví dụ Trigger INSERT: Kiểm tra dữ liệu chèn vào bảng nhân viên có lương phải lớn hơn 5000

```
--Kiểm tra dữ liệu chèn vào bảng nhân viên có lương phải lớn hơn 5000
CREATE TRIGGER CheckLuong_NV ON NhanVien FOR INSERT AS
IF (SELECT LUONG From inserted) < 5000
BEGIN
PRINT 'Tien luong toi thieu phai lon hon 5000.'
ROLLBACK TRANSACTION
END
```

```
INSERT INTO NHANVIEN
VALUES ('Phan', 'Viet', 'The', '115',
'1967-02-01 00:00:00.000', 'hcm', 'nam', 4000, '005', 4)
```

Messages

Tien luong toi thieu phai lon hon 5000.

Msg 3609, Level 16, State 1, Line 9

The transaction ended in the trigger. The batch has been aborted.

❑ Trigger UPDATE

- ❖ Trigger UPDATE sao chép bản ghi gốc vào bảng Deleted và bản ghi mới vào bảng Inserted
- ❖ Nếu các giá trị mới là hợp lệ thì bản ghi từ bảng Inserted sẽ được sao chép vào bảng dữ liệu
- ❖ Trigger UPDATE được tạo ra bằng cách sử dụng từ khóa UPDATE trong câu lệnh CREATE TRIGGER và ALTER TRIGGER.
- ❖ Cú pháp tương tự trigger insert

```
CREATE TRIGGER tên_trigger ON tên_bảng  
FOR {DELETE, INSERT, UPDATE}  
AS  
    câu_lệnh_sql
```


- ❑ Ví dụ trigger cập nhật lương cho nhân viên, quy định lương > 5000

```
CREATE TRIGGER Luong_NV ON NhanVien FOR UPDATE AS  
IF (SELECT LUONG From inserted) < 5000  
BEGIN  
PRINT 'Tien lương tối thiểu phải lớn hơn 5000.'  
ROLLBACK TRANSACTION  
END
```

```
UPDATE NHANVIEN SET LUONG=2000 WHERE MANV like '005'
```



Messages

Tien lương tối thiểu phải lớn hơn 5000.

Msg 3609, Level 16, State 1, Line 21

The transaction ended in the trigger. The batch has been aborted.

❑ Trigger Delete

- ❖ Ta có thể tạo ra trigger DELETE để hạn chế người dùng không xóa một bản ghi cụ thể trong bảng
- ❖ Khi người dùng cố gắng xóa bản ghi:
 - Bản ghi bị xóa khỏi bảng trigger và chèn vào bảng Deleted
 - Kiểm tra các ràng buộc liên quan khi xóa
 - Nếu có ràng buộc trên bản ghi để ngăn chặn việc xóa, trigger DELETE sẽ hiển thị thông báo lỗi
 - Bản ghi đã xóa được lưu trữ trong bảng Deleted được sao chép ngược lại bảng trigger.

```
CREATE TRIGGER <trigger_name> ON <table_name> [WITH ENCRYPTION] FOR DELETE AS  
<sql_statement>
```

- ❑ Ví dụ tạo trigger Delete không cho phép xóa nhân viên có mã 005

```
CREATE TRIGGER XoaNV
ON NhanVien
FOR DELETE
AS
IF '005' IN (SELECT MANV FROM deleted)
BEGIN
PRINT 'Không thể xóa bản ghi này.'
ROLLBACK TRANSACTION
END
```

```
DELETE FROM NHANVIEN WHERE MANV like '005'
```

Không thể xóa bản ghi này.

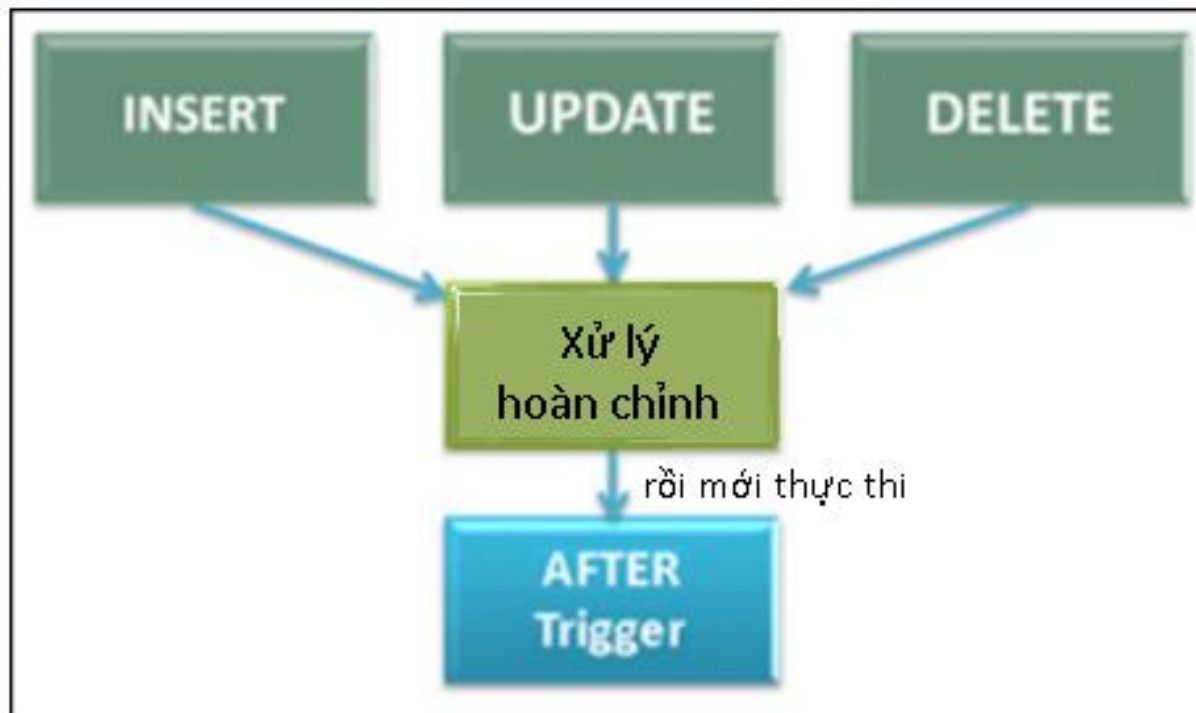


- ❖ Viết trigger rang buộc quy định mức lương của nhân viên không được <5000 cho thao tác thêm và cập nhật bảng nhân viên.
- ❖ Viết trigger rang buộc không được xóa nhân viên ở TP HCM



PHẦN 2

- ❑ Trigger AFTER thực thi khi hoàn thành các hoạt động INSERT, UPDATE và DELETE.
- ❑ Trigger AFTER chỉ có thể được tạo ra trên các bảng.



❑ Cú pháp:

```
CREATE TRIGGER <trigger_name>
ON <table_name>
[WITH ENCRYPTION]
{ FOR | AFTER }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS <sql_statement>
```

❑ Viết trigger đếm số lượng nhân viên bị xóa khi có câu lệnh xóa trên bảng nhân viên

```
CREATE TRIGGER Xoa_NV
ON Nhanvien
AFTER DELETE
AS
BEGIN
DECLARE @num nchar;
SELECT @num = COUNT(*) FROM deleted
PRINT N'Số lượng nhân viên đã xóa = ' + @num
END
```

```
DELETE FROM NHANVIEN WHERE MANV='017'
```

Messages

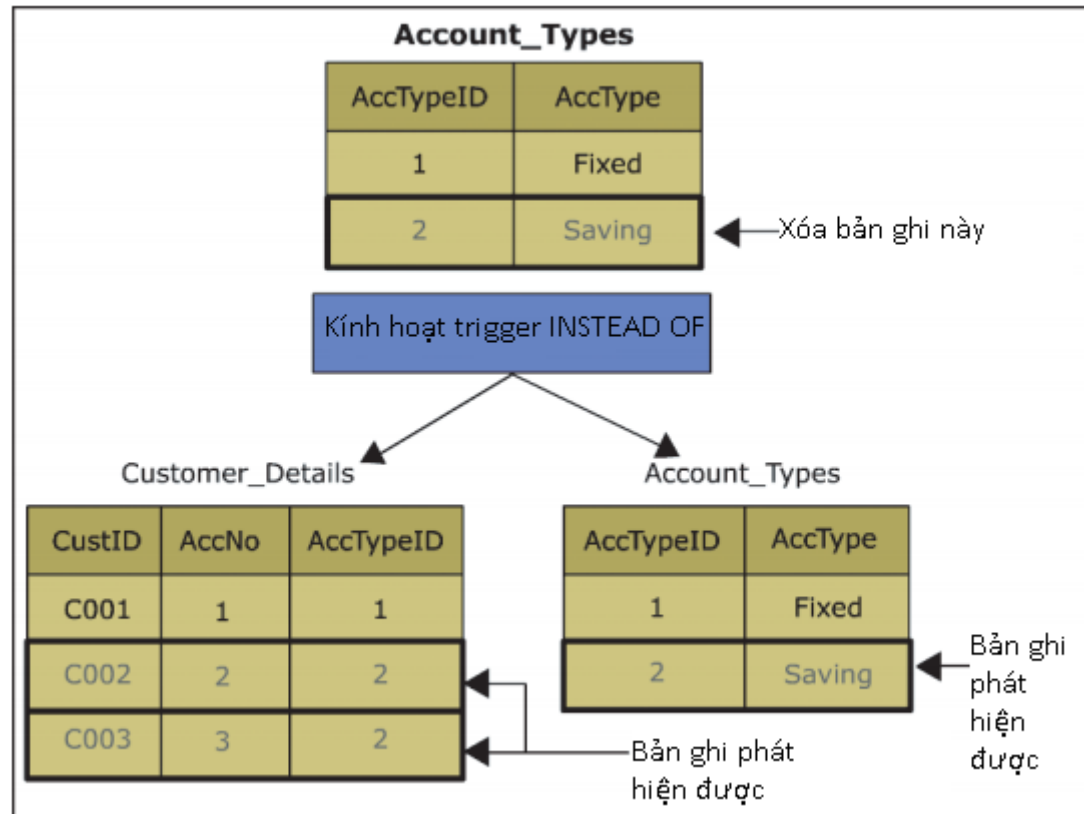
Số lượng nhân viên đã xóa = 0

(0 rows affected)



❖ Viết trigger đếm số lượng nhân viên bị xóa khi thực hiện xóa các nhân viên ở TP HCM.

- ❑ Trigger INSTEAD OF được thực thi thay thế cho các hoạt động INSERT, UPDATE hoặc DELETE.
- ❑ Các trigger INSTEAD OF có thể được tạo ra trên các bảng cũng như khung nhìn.
- ❑ VD: trigger thực hiện xóa luôn các dòng dữ liệu có liên quan ở các bảng Customer_Details và Account_Types



❑ Cú pháp:

```
CREATE TRIGGER <trigger_name>
ON { <table_name> | <view_name> }
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS <sql_statement>
```

❑ Vd xóa nhân viên có mã 017, các thân nhân của nhân viên này tự động bị xóa theo trên bảng thân nhân

```
--Tạo trigger xóa các thân nhân có liên
CREATE TRIGGER Delete_NV_ThanNhan
ON NhanVien
INSTEAD OF DELETE
AS
DELETE FROM NHANVIEN WHERE MANV like '017'
BEGIN
DELETE FROM THANNNHAN WHERE MA_NVIENT IN
(SELECT MA_NVIENT FROM deleted)
DELETE FROM NHANVIEN WHERE MANV IN
(SELECT MANV FROM deleted)
END
```



- ❖ Viết trigger xóa các thân nhân có liên quan đến nhân viên khi nhân viên bị xóa trong bảng nhân viên.

- ☑ Tổng quan trigger
 - ❖ Giới thiệu
 - ❖ Các loại trigger
 - ❖ Các trigger DML
- ☑ Làm việc với trigger
 - ❖ Trigger AFTER
 - ❖ Trigger INSTEAD OF



thank
you!