



HỆ QUẢN TRỊ CSDL

Số tc: 3 (2 Lec + 1 Lab)

Giảng viên: Lương Thị Thu Phương

Phone: 0935 512 766

Email: luongthuphuongitdtu@gmail.com



Chương 3: VIEW – PROCEDURE - TRIGGER

VIEW

Một khung nhìn VIEW là một bảng ảo trong cơ sở dữ liệu có nội dung được định nghĩa thông qua một câu lệnh SQL nào đó. Một **VIEW** bao gồm các hàng và cột giống như một bảng thực.

VIEW không được xem là một cấu trúc lưu trữ dữ liệu tồn tại trong cơ sở dữ liệu.



Chương 3: VIEW – PROCEDURE - TRIGGER

VIEW

CREATE VIEW <tên_view> AS

SELECT cột1, cột2,

FROM <tên_bảng|ds bảng>

WHERE [điều_kiện];





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

1. Stored Procedure là gì?

Procedure (Thủ tục) là một chương trình trong cơ sở dữ liệu gồm nhiều câu lệnh mà ta có thể lưu lại để sử dụng cho những lần sau.

Procedure là tập hợp một hoặc nhiều câu lệnh T-SQL thành một nhóm đơn vị xử lý logic và được lưu trữ trên Database Server.





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

2. Tạo Stored Procedure trong SQL Server

Create Procedure <Pro_Name>

AS

BEGIN

<SQL Statement>

END;





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

Ví dụ 1:

```
CREATE PROCEDURE PRO_NHANVIEN
```

NHANVIEN

```
AS
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM NHANVIEN
```

```
    ORDER BY MANV
```

```
END;
```

MANV	TENNV	LUONG
NV01	ANH	5000000
NV02	HÔNG	6000000
NV03	HẠNH	4500000
NV04	LINH	3500000
NV05	VY	6500000

Commands completed successfully.



Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

Ví dụ 2:

```
CREATE PROCEDURE PRO_NHANVIEN
```

NHANVIEN

```
AS
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM NHANVIEN
```

```
    WHERE LUONG >= 5000000
```

```
    ORDER BY MANV
```

```
END;
```

MANV	TENNV	LUONG
NV01	ANH	5000000
NV02	HÔNG	6000000
NV05	VY	6500000

Commands completed successfully.



Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

3. Gọi Stored Procedure trong SQL Server

EXECUTE PRO_Name;

Hoặc

EXEC PRO_Name;





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

4. Thay đổi Stored Procedure trong SQL Server

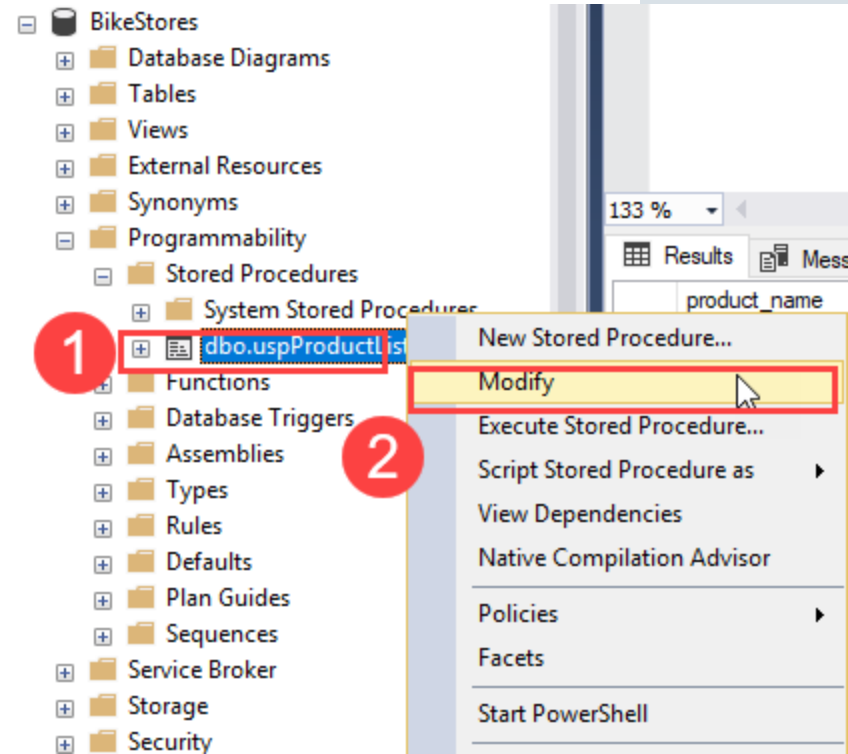
ALTER PROCEDURE <Pro_Name>

AS

BEGIN

.....

END;





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

5. Xóa Stored Procedure trong SQL Server

`DROP PROCEDURE <Pro_Name>;`

Hoặc

`DROP PROC <Pro_Name>;`





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

Tạo Procedure với tham số đầu vào

```
Create Procedure <Procedure_Name>  
(@param_name_1      param_type_1,  
  @param_name_2      param_type_2, ...,  
  @param_name_n      param_type_n)  
As  
Begin  
  <SQL Statement>  
END;
```





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

VD1: Tạo stored procedure với tham số đầu vào

```
CREATE PROCEDURE SELECT_NHANVIEN
```

```
(@LUONG INT)
```

```
AS
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM NHANVIEN
```

```
    WHERE LUONG >= @LUONG
```

```
END;
```

NHANVIEN

MANV	TENNV	LUONG
NV01	ANH	5000000
NV02	HÔNG	6000000
NV03	HẠNH	4500000
NV04	LINH	3500000
NV05	VY	6500000

```
EXEC SELECT_NHANVIEN 5000000
```



Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

Tạo Procedure với tham số đầu vào có giá trị mặc định

```
CREATE PROCEDURE SELECT_NHANVIEN  
  (@LUONG INT = 5000000)  
AS  
BEGIN  
    SELECT *  
    FROM NHANVIEN  
    WHERE LUONG >= @LUONG  
END;
```

EXEC SELECT_NHANVIEN 5000000





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

Tạo Procedure với tham số OUTPUT

```
CREATE PROCEDURE SELECT_NHANVIEN  
(@LUONG INT = 5000000, @NHANVIEN_COUNT INT OUTPUT)  
AS  
BEGIN  
    SELECT *  
    FROM NHANVIEN  
    WHERE LUONG >= @LUONG  
    SELECT @NHANVIEN_COUNT = @@ROWCOUNT;  
END;
```





Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

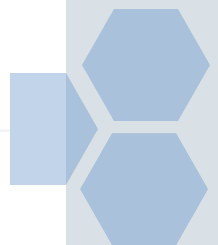
Thực thi Procedure với tham số OUTPUT

```
DECLARE @COUNT INT;  
EXEC SELECT_NHANVIEN  
@LUONG = 5000000  
@NHANVIEN_COUNT = @COUNT;  
SELECT @COUNT AS 'TOTAL';
```

Output:

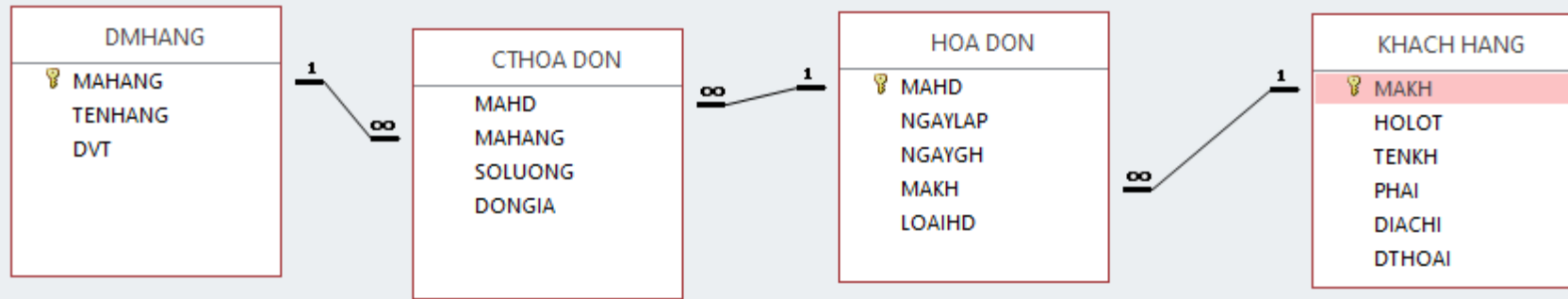
Total

4





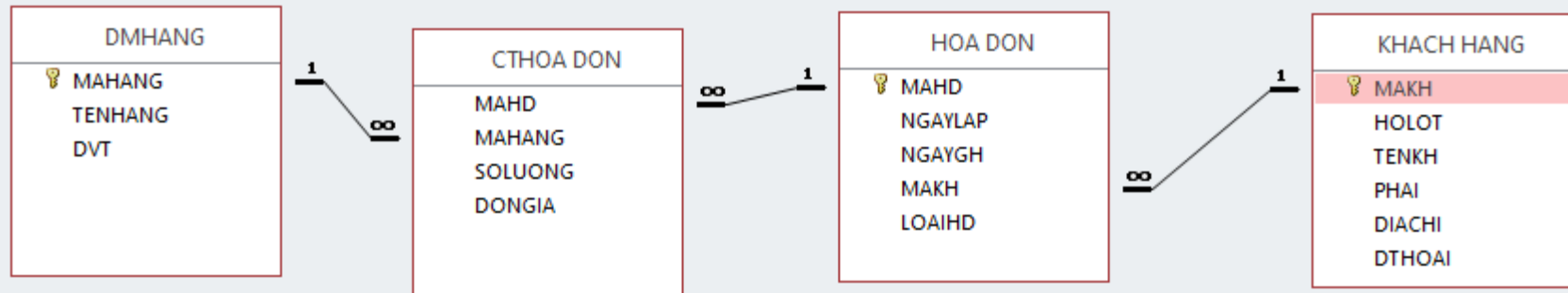
Chương 3: VIEW – PROCEDURE - TRIGGER



1. Tạo View_ThanhTien như sau: MAHD, NGAYLAP, NGAYGH, TENHANG, SOLUONG, DONGIA, THANH TIEN.
2. Lập ds 3 hóa đơn có Thành tiền cao nhất? (lấy nguồn từ View).
3. Tạo Procedure hiển thị các khách hàng có địa chỉ Quảng Nam, Đà Nẵng đặt hàng vào tháng 4. (tự lấy thông tin phù hợp).
4. Tạo Procedure hiển thị các mặt hàng có số lượng ≥ 10 có ngày giao hàng là ngày 15. (tự lấy thông tin phù hợp).
5. Tạo Procedure hiển thị số các hóa đơn là M của các khách hàng không phải ở Đà Nẵng đã mua hàng vào tháng 2 (tự lấy thông tin phù hợp).



Chương 3: VIEW – PROCEDURE - TRIGGER



6. Tạo Procedure có tham số đầu vào là Số lượng. Hiển thị các hóa đơn có số lượng ≥ 5 : MAHD, NGAYLAP, NGAYGIAO, TENHANG, SOLUONG.

7. Tạo Procedure có tham số đầu vào là NGAYLAP. Hiển thị các hóa đơn có ngày lập trong tháng 2: MAHD, NGAYLAP, NGAYGIAO, TENHANG, SOLUONG, DONGIA, THANH TIEN.

8. Tạo Procedure có tham số đầu ra cho biết số mặt hàng trên mỗi hóa đơn (tự lấy thông tin phù hợp).

9. Tạo Procedure có tham số đầu ra cho biết số mặt hàng trên mỗi hóa đơn có số lượng ≥ 10 (tự lấy thông tin phù hợp).



Chương 3: VIEW – PROCEDURE - TRIGGER

PROCEDURE

Tạo Procedure nhập thêm dữ liệu vào Table

Tạo Procedure Update dữ liệu của Table





Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER

Trigger là gì ?

Trigger là một stored procedure không có tham số. Trigger thực thi một cách tự động khi một trong ba câu lệnh **Insert**, **Update**, **Delete** làm thay đổi dữ liệu trên bảng có chứa trigger





Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER

Tạo Trigger

```
CREATE TRIGGER <Trigger_Name>  
ON <Table_Name>  
FOR {INSERT | DELETE | UPDATE}  
AS  
    BEGIN  
        <SQL Statement>  
    END
```





Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER

- Trigger thường được sử dụng để kiểm tra ràng buộc (check constraints) trên nhiều quan hệ hoặc trên nhiều dòng của bảng.
- Vai trò của Trigger trong CSDL là tính nhất quán dữ liệu.





Chương 3: VIEW – PROCEDURE - TRIGGER

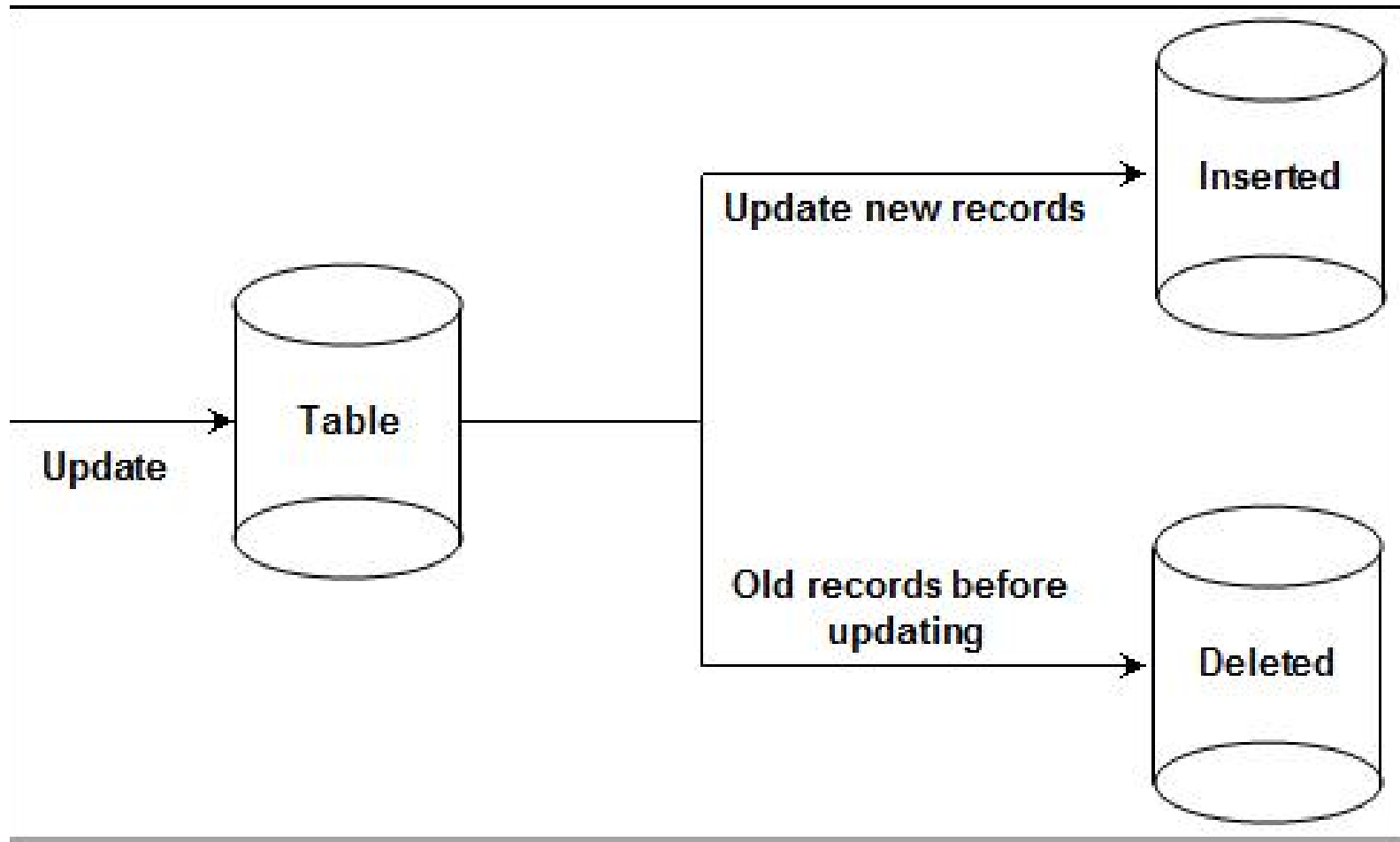
TRIGGER

Khi trigger được thực hiện, SQL tự động tạo ra 2 bảng tạm với cùng cấu trúc với bảng chứa trigger.

- **INSERTED** chứa dữ liệu mới khi chúng ta thực thi câu lệnh Insert hoặc câu lệnh Update.
- **DELETED** chứa những bản ghi bị xoá khi chúng ta thực thi câu lệnh Delete hoặc chứa dữ liệu cũ khi chúng ta thực thi câu lệnh Update.

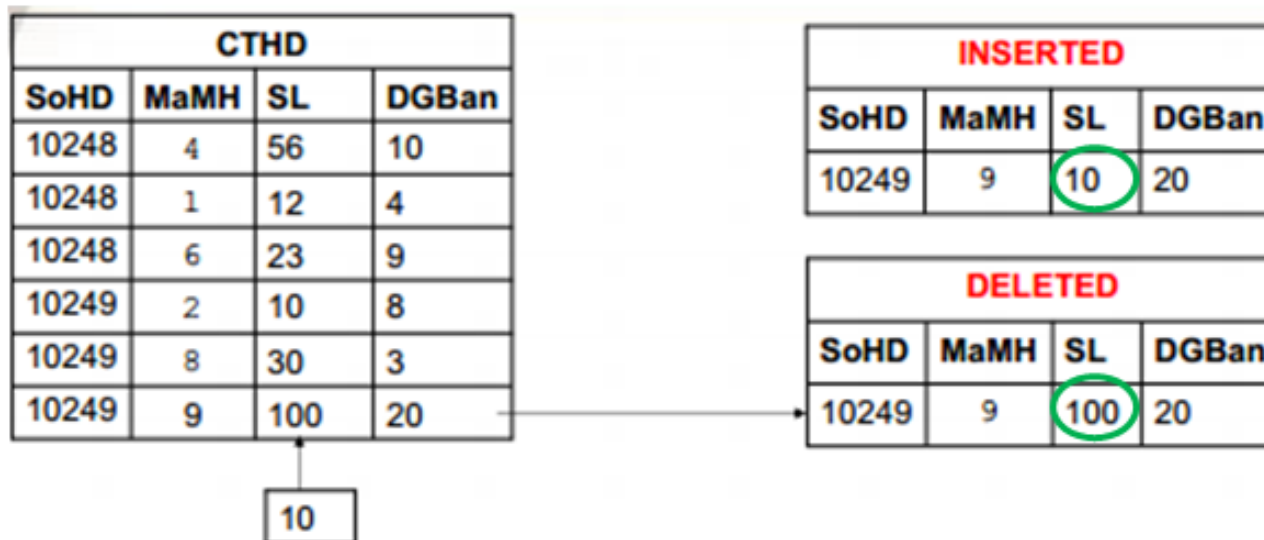
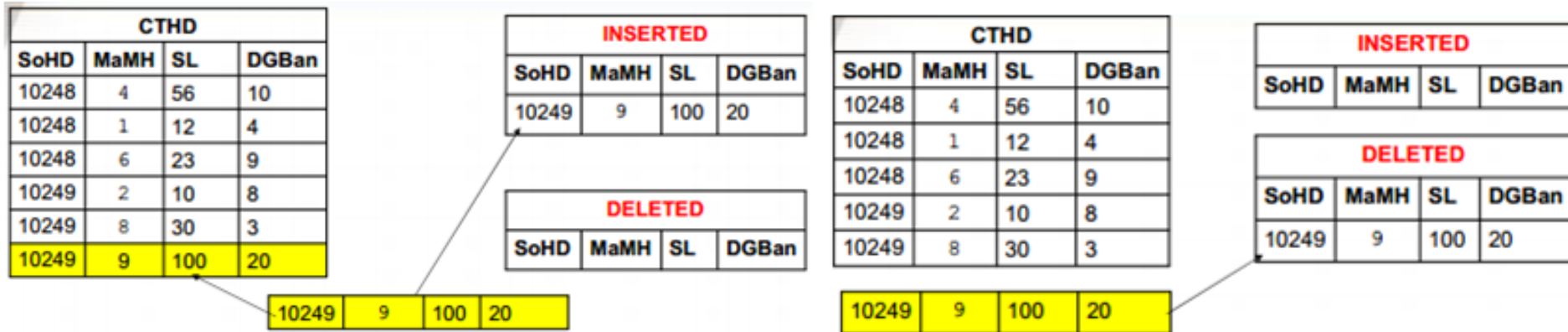


Chương 3: VIEW – PROCEDURE - TRIGGER





Chương 3: VIEW – PROCEDURE - TRIGGER

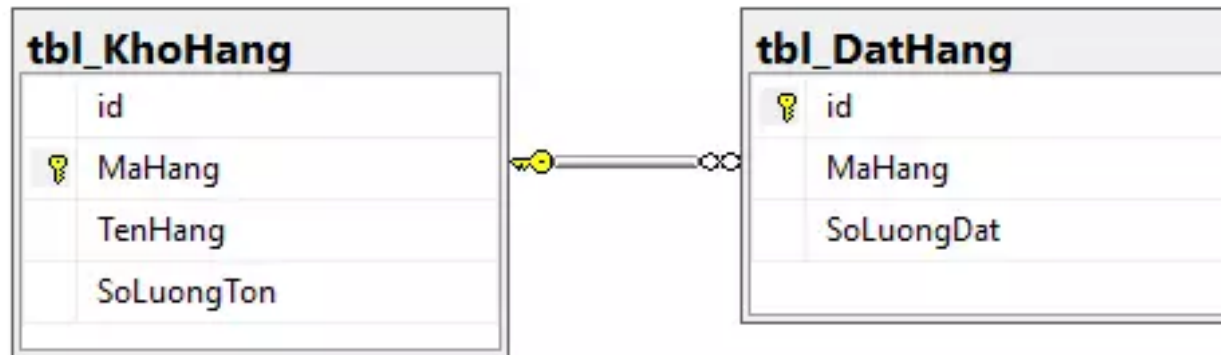




Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER

Ví dụ:



Khi có khách đặt hàng thì tự động cập nhật số lượng tồn trong bảng kho hàng.





Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER

Giải pháp:

Khi khách đặt hàng ta chỉ có 3 loại thao tác chính với CSDL là :

Insert, Delete, Update → // Tạo 3 trigger.

- Khách **đặt hàng**:

Số lượng tồn trong kho = Số lượng tồn - Số lượt đặt

- Khách **hủy** không đặt hàng nữa:

Số lượng tồn trong kho = Số lượng tồn + Số lượt đặt

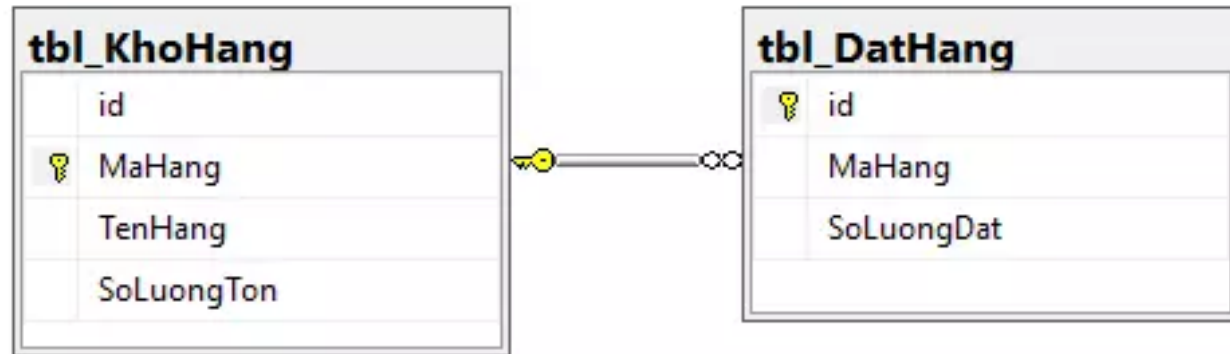
- Khách **cập nhật** số lượng đặt





Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER



Thực hiện:

1. Nhập dữ liệu cho bảng KhoHang, DatHang không nhập.
2. Giả sử đặt hàng với SoluongDat: 3 SP mặt hàng MH01 → **Insert**
3. Hủy đặt hàng với MH01 → **Delete**
4. Cập nhật lại đặt hàng thêm 5 SP với MH01 → **Update**





Chương 3: VIEW – PROCEDURE - TRIGGER

*//*Đặt hàng*

```
CREATE TRIGGER trg_DatHang ON tbl_DatHang
FOR INSERT
AS
BEGIN
    UPDATE tbl_KhoHang
    SET SoLuongTon = SoLuongTon - (
        SELECT SoLuongDat
        FROM inserted
        WHERE MaHang = tbl_KhoHang.MaHang )
    FROM tbl_KhoHang
    JOIN inserted ON tbl_KhoHang.MaHang = inserted.MaHang
END
```





Chương 3: VIEW – PROCEDURE - TRIGGER

*/** Hủy đặt hàng*

```
CREATE TRIGGER trg_HuyDatHang ON tbl_DatHang
FOR DELETE
AS
BEGIN
    UPDATE tbl_KhoHang
    SET SoLuongTon = SoLuongTon + (
        SELECT SoLuongDat
        FROM deleted
        WHERE MaHang = tbl_KhoHang.MaHang )
    FROM tbl_KhoHang
    JOIN deleted ON tbl_KhoHang.MaHang = deleted.MaHang
END
```





Chương 3: VIEW – PROCEDURE - TRIGGER

*/** cập nhật đặt hàng*

```
CREATE TRIGGER trg_CapnhatDatHang ON tbl_DatHang
```

```
FOR UPDATE
```

```
AS
```

```
BEGIN
```

```
UPDATE tbl_KhoHang
```

```
SET SoLuongTon = SoLuongTon -
```

```
(SELECT SoLuongDat FROM inserted WHERE MaHang = tbl_KhoHang.MaHang )
```

```
+ (SELECT SoLuongDat FROM deleted WHERE MaHang = tbl_KhoHang.MaHang )
```

```
FROM tbl_KhoHang
```

```
JOIN deleted ON tbl_KhoHang.MaHang = deleted.MaHang
```

```
END
```





Chương 3: VIEW – PROCEDURE - TRIGGER

TRIGGER

