

## **BÀI THỰC HÀNH 4**

### **Stored Procedure (SP)**

---

#### **1. Mục tiêu**

Sau khi thực hành bài này, sinh viên có thể:

- Hiểu cách khai báo một SP
- Biết vận dụng các câu lệnh đơn giản để lập trình giải quyết bài toán với SP.

#### **2. Yêu cầu**

- Sinh viên đọc kỹ các nội dung lý thuyết đã học về SP.
- Các bài tập phải được giải quyết thông qua lập trình câu lệnh.

#### **3. Bài tập**

##### **• Bài tập 1: thực hiện với các thủ tục hệ thống**

- 1.** Sử dụng thủ tục “sp\_help” để xem thông tin về CSDL và bảng.

##### **Hướng dẫn:**

1. Xem thông tin CSDL: **execute sp\_help**
2. Xem thông tin bảng titles trong CSDL Pubs: **execute sp\_help titles**

##### **• Bài tập 2:**

1. Viết thủ tục tính số ngày của một tháng trong năm. Với yêu cầu:
  - *Input: tháng năm dạng tháng/năm (ví dụ: 2/2009)*
  - *Output: Số ngày của tháng (ví dụ: 28)*
2. Thay vì dùng thủ tục, định nghĩa 1 hàm để tính số ngày của 1 tháng trong năm.

##### **• Bài tập 3: CSDL QLHangHoa**

1. Viết thủ tục tính số lượng tồn của 1 mặt hàng của 1 đại lý cụ thể.

```
Create Procedure TonKho
    @STT_DL int, @Ma_HAng nvarchar(3),
    @TongMua int Output, @TongBan int Output, @Ton int Output
As
Begin
    Select @TongMua=Sum(SoLg_Mua)
    From Mua
    Where STT_DL=@STT_DL and Ma_HAng=@Ma_Hang
```

```

If @TongMua Is NULL Return

Select @TongBan=Sum(SoLg_Ban)
From Ban
Where STT_DL=@STT_DL and Ma_Hang=@Ma_Hang

If @TongBan Is NULL
    Set @TongBan = 0

Set @Ton = @TongMua-@TongBan
End

```

### **Gọi thực thi thủ tục:**

```

Declare @Mua int, @Ban int, @Ton int
Execute TonKho 2, '002', @Mua Output, @Ban Output, @Ton Output
print 'Tong mua: ' + Cast(@Mua as VArchar(20)) +
      '. Tong ban ' + Cast(@Ban as varchar(20)) +
      '. Ton: ' + Cast(@Ton as varchar(20))

```

2. Viết thủ tục bán 1 mặt hàng của một đại lý cụ thể với số lượng và đơn giá bán cho trước, ngày bán là ngày hiện hành

```

Create Procedure BanHang
    @STT_DL int, @Ma_Hang nvarchar(3), @SoLgBan int, @DonGia int
As
Begin
    Declare @TongMua int, @TongBan int
    Select @TongMua=Sum(SoLg_Mua)
    From Mua
    Where STT_DL=@STT_DL and Ma_HAng=@Ma_Hang

    If @TongMua Is NULL
    Begin
        print 'Mat hang ' + @Ma_Hang + ' khong co' + ' o dai ly'
        Return
    End
End

```

```

Select @TongBan=Sum(SoLg_Ban)
From Ban
Where STT_DL=@STT_DL and Ma_Hang=@Ma_Hang

if @TongBan Is Null
    Set @TongBan=0
if @TongMua >= @TongBan + @SoLgBan
    Insert Into Ban Values(@Ma_Hang, @STT_DL, getDate(),
        @SoLgBan, @DonGia)
else
    print 'Khong du hang de ban'
End

```

### Thực thi thủ tục:

Execute BanHang 2, '002', 160, 12

3. Viết thủ tục cho phép tính tồn kho của mỗi mặt hàng của một đại lý cụ thể (STT\_DL là tham số).
4. Viết thủ tục cho phép tính tồn kho của mỗi mặt hàng của mỗi đại lý.
5. Giả sử các khóa ngoại của bảng BAN và MUA có thuộc loại **On Delete No Action**. Viết thủ tục cho phép xóa một hàng hóa cùng các thông tin liên quan tới hàng hóa này ở bảng BAN và MUA với mã hàng là tham số.

**Gợi ý:**

- Thực hiện xóa ở bảng BAN, MUA
- Thực hiện xóa ở bảng HANGHOA

### • Bài tập 5:

Tạo bảng sau:

```

Create table nhanvien
(
    manv int not null primary key,
    hoten varchar(30) not null,
    diachi varchar(30) not null)

```

1. Viết thủ tục tên “insert\_nhanvien” cho phép thêm một mẫu tin nhân viên vào bảng nhanvien .

2. Viết đoạn lệnh thêm một mẫu tin nhân viên với nội dung sau:

1, ‘Nguyen Van Thanh’, ‘01 Ly Tu Trong, NK– TPCT’

Sau đó hiển thị bảng nhân viên để kiểm tra kết quả.

3. Viết thủ tục cập nhật địa chỉ của nhân viên

- *Input: mã nhân viên và địa chỉ đúng*

4. Viết đoạn lệnh cập nhật địa chỉ của nhân viên có manv=1 và địa chỉ đúng là

‘10 Ly Tu Trong, Q. NK – TPCT’

Sau đó hiển thị bảng nhân viên để kiểm tra kết quả.

#### • **Bài tập 5: Sử dụng CSDL Northwind**

1. Viết thủ tục/hàm tính tiền hóa đơn với

- *Input: ORDERID*

- *Output: Trị giá hóa đơn tương ứng*

2. Hướng dẫn:

##### + **Cách 1:**

```
Create Proc CalulateMoney_Order
@Order_ID char(5)
AS
select sum(UnitPrice*Quantity-
          UnitPrice*Quantity*Discount) as Total_Price
from [Order Details]
where OrderId=@Order_ID
-----
Exec CalulateMoney_Order '10249'
```

##### + **Cách 2:**

```
create function dbo.CalulateMoney(@Order_ID char(5))
returns table
as
return (select sum(UnitPrice*Quantity-
          UnitPrice*Quantity*Discount) as Total_Price
```

```

        from [Order Details]
        where OrderId=@Order_ID)
-----
select *
from dbo.CalulateMoney('10249')
+ Cách 3:
create function dbo.Calulate(@Order_ID char(5))
returns money
as
Begin
    Declare @a money
    select @a= sum(UnitPrice*Quantity -
                    UnitPrice*Quantity*Discount)
    from [Order Details]
    where OrderId=@Order_ID
    return @a
End
-----
select dbo.Calulate('10249')

```

### • Bài tập 6

1. Tạo bảng Account chứa thông tin về tài khoản của 1 người với mã số tài khoản là khóa chính (int); ngoài ra còn có tên tài khoản (varchar(30))
2. Biết rằng số tài khoản (account number) gồm 9 chữ số chẳng hạn 972428577. Viết một hàm kiểm tra số tài khoản này có hợp lệ theo nguyên tắc kiểm tra chia dư 11 ('modulo 11-check').

Thí dụ:

972428577 là hợp lệ vì:

$$(9*9 + 8*7 + 7*2 + 6*4 + 5*2 + 4*8 + 3*5 + 2*7 + 1*7) \% 11 = 0$$

3. Sử dụng hàm này để định nghĩa ràng buộc toàn vẹn: Số tài khoản phải thỏa mãn nguyên tắc 'chia dư 11'

**Gợi ý:** Dùng 1 check constraint