

STORED PROCEDURE

Nội dung

1. Tổng quan
2. Bài tập

I. TỔNG QUAN

1. Stored procedure là gì?

- Stored procedure (SP) được gọi là thủ tục lưu trữ.
- Một thủ tục là một đối tượng trong cơ sở dữ liệu bao gồm một tập nhiều câu lệnh SQL được nhóm lại với nhau thành một nhóm với những khả năng sau:
 - Có thể chứa các cấu trúc điều khiển
 - Có thể sử dụng biến trong thủ tục như trong ngôn ngữ lập trình
 - Nhận tham số truyền vào, có khả năng trả về giá trị thông qua tham số.
 - Một tập các câu lệnh SQL được kết hợp lại với nhau thành một khối lệnh bên trong một thủ tục.

I. TỔNG QUAN

- Sử dụng các thủ tục lưu trữ mang lại các lợi ích sau:
 - Đơn giản hoá các thao tác.
 - Thực thi thủ tục nhanh hơn nhiều so với việc phải thực hiện một tập rời rạc các câu lệnh SQL tương đương theo cách thông thường.
 - Giảm thiểu sự lưu thông trên mạng.
 - Tăng khả năng bảo mật đối với hệ thống

I. TỔNG QUAN

2. Khởi tạo Stored procedure

2.1. Cú pháp:

```
CREATE PROCEDURE tên_thủ_tục [(danh_sách_tham_số)]  
[WITH RECOMPILE | ENCRYPTION]  
AS  
BEGIN  
    Các_câu_lệnh_của_thủ_tục  
END
```

I. TỔNG QUAN

Trong đó:

- tên thủ tục: tên của thủ tục cần tạo. Tên phải tuân theo qui tắc định danh và không được vượt quá 128 ký tự.

Ví dụ: usp_themnhanvien

- danh sách tham số: các tham số của thủ tục được khai báo ngay sau tên thủ tục và nếu thủ tục có nhiều tham số thì các khai báo phân cách nhau bởi dấu phẩy. Khai báo của mỗi một tham số tối thiểu phải bao gồm hai phần:

- tên tham số được bắt đầu bởi dấu @.
- kiểu dữ liệu của tham số

Ví dụ: @mamonhoc nvarchar(10)

I. TỔNG QUAN

Trong đó:

- RECOMPILE : Thông thường, thủ tục sẽ được phân tích, tối ưu và dịch sẵn ở lần gọi đầu tiên. Nếu tùy chọn WITH RECOMPILE được chỉ định, thủ tục sẽ được dịch lại mỗi khi được gọi.
- ENCRYPTION : Thủ tục sẽ được mã hoá nếu tùy chọn WITH ENCRYPTION được chỉ định. Nếu thủ tục đã được mã hoá, ta không thể xem được nội dung của thủ tục.
- các_câu_lệnh_của_thủ_tục: Tập hợp các câu lệnh sử dụng trong nội dung thủ tục

I. TỔNG QUAN

2. Khởi tạo Stored procedure

ví dụ: Tạo SP tìm các nhân viên làm việc ở phòng bất kỳ do người dùng nhập vào

```
CREATE PROC usp_timnhanvienphong @phong int
```

```
AS
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM NHANVIEN NV
```

```
    WHERE NV.phg = @phong
```

```
END
```


I. TỔNG QUAN

2. Khởi tạo Stored procedure

2.2. Cú pháp khởi tạo với tham số mặc định:

- Các tham số được khai báo trong thủ tục có thể nhận các giá trị mặc định.
- Giá trị mặc định sẽ được gán cho tham số trong trường hợp không truyền đối số cho tham số khi có lời gọi đến thủ tục.
- Tham số với giá trị mặc định được khai báo theo cú pháp như sau:
@tên_tham_số kiểu_dữ_liệu = giá_trị_mặc_định

I. TỔNG QUAN

ví dụ:

```
CREATE PROCEDURE usp_tinh tong_defaultparam
```

```
@a int = 10,
```

```
@b int = 5
```

```
AS
```

```
begin
```

```
    declare @tong int
```

```
    set @tong = @a + @b
```

```
    print @tong
```

```
end
```

I. TỔNG QUAN

3. Thực thi Stored procedure

Cú pháp:

EXECUTE tên_thủ_tục [danh_sách_các_đối_số]

Ví dụ:

EXEC usp_timnhanvienphong 5

I. TỔNG QUAN

3. Thực thi Stored procedure

Thứ tự của các đối số được truyền cho thủ tục có thể không cần phải tuân theo thứ tự của các tham số như khi định nghĩa thủ tục nếu tất cả các đối số được viết dưới dạng:

@tên_tham_số = giá_trị

I. TỔNG QUAN

ví dụ:

Lời gọi thủ tục có thể viết như sau:

```
EXECUTE usp_LenDanhSachDiem @malop='C24102',  
                                     @tenmonhoc='Cơ sở dữ liệu',  
                                     @mamonhoc='TI-005',  
                                     @sodvht=5
```

I. TỔNG QUAN

ví dụ:

- Gọi thủ tục với tham số mặc định:

EXEC usp_tinh tong_defaultparam

- Gọi thủ tục với tham số ghi đè tham số mặc định

EXEC usp_tinh tong_defaultparam 10, 10

I. TỔNG QUAN

4. Sử dụng biến trong Stored procedure

Ngoài những tham số được truyền cho thủ tục, bên trong thủ tục còn có thể sử dụng các biến nhằm lưu giữ các giá trị tính toán được hoặc truy xuất được từ cơ sở dữ liệu.

Các biến trong thủ tục được khai báo bằng từ khoá DECLARE theo cú pháp như sau:

DECLARE @tên_biến kiểu_dữ_liệu

I. TỔNG QUAN

4. Sử dụng biến trong Stored procedure

Ví dụ:

```
CREATE PROCEDURE usp_tinh tong
```

```
@a int,
```

```
@b int
```

```
AS
```

```
begin
```

```
    declare @tong int
```

```
    set @tong = @a + @b
```

```
    print @tong
```

```
end
```


I. TỔNG QUAN

5. Giá trị trả về của tham số trong Stored procedure

Nếu đối số truyền cho thủ tục khi có lời gọi đến thủ tục là biến, những thay đổi giá trị của biến trong thủ tục sẽ không được giữ lại khi kết thúc quá trình thực hiện thủ tục.

Trong trường hợp cần phải giữ lại giá trị của đối số sau khi kết thúc thủ tục, ta phải khai báo tham số của thủ tục theo cú pháp như sau:

@tên_tham_số kiểu_dữ_liệu OUTPUT

hoặc: **@tên_tham_số kiểu_dữ_liệu OUT**

và trong lời gọi thủ tục, sau đối số được truyền cho thủ tục, ta cũng phải chỉ định thêm từ khoá OUTPUT (hoặc OUT)

I. TỔNG QUAN

Ví dụ:

```
CREATE PROCEDURE usp_tinhhtongoutput
```

```
@a int,
```

```
@b int,
```

```
@tong int output
```

```
AS
```

```
begin
```

```
    set @tong = @a + @b
```

```
end
```

I. TỔNG QUAN

Lời gọi thủ tục được thực hiện như sau:

declare @tongout int -- *khai báo biến nhận kết quả trả về*

EXEC usp_tinhtongoutput 5,10, @tongout output -- *gọi thủ tục*

print @tongout -- *in kết quả (có thể thay **print** bằng **select**)*

I. TỔNG QUAN

6. Sửa Stored procedure

Cú pháp:

ALTER PROCEDURE tên_thủ_tục [(danh_sách_tham_số)]

[**WITH RECOMPILE** | **ENCRYPTION**]

AS

BEGIN

Các_câu_lệnh_của_thủ_tục

END

I. TỔNG QUAN

7. Xóa Stored procedure

Cú pháp:

DROP PROCEDURE tên_thủ_tục

Khi xoá một thủ tục, tất cả các quyền đã cấp cho người sử dụng trên thủ tục đó cũng đồng thời bị xoá bỏ.

Do đó, nếu tạo lại thủ tục, ta phải tiến hành cấp phát lại các quyền trên thủ tục đó.

II. MỘT SỐ VÍ DỤ

--1. Tìm các nhân viên làm việc ở phòng bất kỳ do người dùng nhập vào

```
CREATE PROC usp_timnhanvienphong
```

```
@phong int
```

```
AS
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM NHANVIEN NV
```

```
    WHERE NV.phg = @phong
```

```
END
```

--gọi SP

```
EXEC usp_timnhanvienphong 5
```

II. MỘT SỐ VÍ DỤ

--2. Kiểm tra xem một nhân viên có trong hệ thống hay không, nếu có thì in ra thông tin nhân viên, ngược lại in ra Không có

```
CREATE PROC usp_kiemtraNV
```

```
@manv char(3)
```

```
AS
```

```
BEGIN
```

```
IF @manv in (SELECT manv FROM NHANVIEN)
```

```
    SELECT *
```

```
    FROM NHANVIEN NV
```

```
    WHERE NV.manv = @manv
```

```
ELSE
```

```
    print N'Mã nhân viên không đúng'
```

```
END
```

II. MỘT SỐ VÍ DỤ

-- 3. Viết thủ tục tìm hóa đơn của một khách hàng theo số điện thoại hoặc địa chỉ.

```
CREATE PROC usp_timkhachhang(@DiaChi nvarchar(40), @SoDienThoai  
nvarchar(14))
```

```
AS
```

```
BEGIN
```

```
    SELECT KH.MaKH, KH.TenKH, HD.MaHD, HD.MaNV, HD.NgayLapHD,  
    HD.NgayGiaoHang
```

```
    FROM HOADON HD, KHACHHANG KH
```

```
    WHERE HD.MaKH=KH.MaKH AND (KH.DienThoai=@SoDienThoai OR  
    KH.DiaChi=@DiaChi)
```

```
END
```

```
EXECUTE usp_timkhachhang N'350 CMT8 P12 Q5', ''
```


III. BÀI TẬP

1. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
 - a. Nhập vào mã nhân viên, kiểm tra xem mã nhân viên đó có tồn tại chưa, nếu có thì in ra thông tin của nhân viên đó, nếu chưa thì in ra thông báo “mã nhân viên không tồn tại”.

```
CREATE PROC usp_inthongtinnhanvien @manv char(3)
```

```
AS
```

```
BEGIN
```

```
-- KIỂM TRA SỰ TỒN TẠI CỦA MÃ NV NHẬP VÀO, NẾU TỒN TẠI MÃ NV THÌ LẤY RA THÔNG TIN
```

```
IF EXISTS (SELECT * FROM NHANVIEN NV WHERE NV.manv = @manv)
```

```
    SELECT * FROM NHANVIEN WHERE MANV = @manv
```

```
ELSE
```

```
    PRINT 'MÃ NHÂN VIÊN KHÔNG TỒN TẠI'
```

```
END
```

```
-- THỰC THI PROC
```

```
EXEC usp_inthongtinnhanvien 444
```

```
EXEC usp_inthongtinnhanvien 123
```

III. BÀI TẬP

1. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
 - b. Nhập vào mã đề án, in ra số lượng nhân viên tham gia đề án đó

--TẠO PROC

CREATE PROC usp_insoluongnhanviendean @mada int

AS

BEGIN

 SELECT COUNT(ma_nvien) FROM PHANCONG WHERE soda
 = @mada

END

--THỰC THI PROC

EXEC usp_insoluongnhanviendean 10

III. BÀI TẬP

1. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
- c. Nhập vào mã nhân viên, in ra số thân nhân của nhân viên đó. Lưu ý: kiểm tra xem mã nhân viên nhập vào, nếu có thì in ra số lượng thân nhân của nhân viên đó, nếu không thì in thông báo mã nhân viên không tồn tại trong hệ thống.

```
CREATE PROC usp_thongkesothannhannv @manv char(3)
AS
BEGIN
    IF @manv IN (SELECT NV.manv FROM NHANVIEN NV)
        SELECT COUNT(*) AS N'SỐ THÂN NHÂN' FROM THANNHAN TN
        WHERE TN.ma_nvien = @manv
    ELSE
        PRINT N'MÃ NHÂN VIÊN KHÔNG TỒN TẠI'
END
-- THỰC THI PROC
EXEC usp_thongkesothannhannv 444;   EXEC usp_thongkesothannhannv 999;
EXEC usp_thongkesothannhannv 123
```

III. BÀI TẬP

1. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
- d. Nhập vào mã nhân viên, in ra số dự án mà nhân viên đó được phân công. Lưu ý: kiểm tra xem mã nhân viên nhập vào, nếu có thì in ra số lượng dự án mà nhân viên đó tham gia, nếu không thì in thông báo mã nhân viên không tồn tại trong hệ thống.

-- TẠO PROC

```
CREATE PROC usp_thongkesoDEANNV @manv char(3)
```

```
AS
```

```
BEGIN
```

```
IF @manv IN (SELECT NV.manv FROM NHANVIEN NV)
```

```
SELECT COUNT(*) AS N'SỐ ĐỀ ÁN THAM GIA' FROM PHANCONG PC WHERE PC.ma_nvien = @manv
```

```
ELSE
```

```
PRINT N'MÃ NHÂN VIÊN KHÔNG TỒN TẠI'
```

```
END
```

-- THỰC THI PROC

```
EXEC usp_thongkesoDEANNV 444; EXEC usp_thongkesoDEANNV 888; EXEC usp_thongkesoDEANNV 123
```

III. BÀI TẬP

1. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
 - e. Nhập vào mã trưởng phòng, in ra thông tin các nhân viên của trưởng phòng đó.

III. BÀI TẬP

1. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
f. Nhập vào mã nhân viên, mã phòng ban. Kiểm tra xem nhân viên đó có thuộc phòng ban được nhập vào hay không.

-- TẠO PROC

```
CREATE PROC usp_thongkeNVtheoPB @manv char(3), @mapb int
AS
BEGIN
```

```
    DECLARE @TEMP INT
```

```
    SELECT @TEMP = COUNT(NV.manv)
```

```
    FROM NHANVIEN NV
```

```
    WHERE NV.manv = @manv AND NV.phg = @mapb
```

```
    IF @TEMP > 0
```

```
        PRINT N'NHÂN VIÊN CÓ MÃ SỐ LÀ ' + @MANV + N' THUỘC PHÒNG ' + CAST(@mapb AS nvarchar(3))
```

```
    ELSE
```

```
        PRINT N'NHÂN VIÊN CÓ MÃ SỐ LÀ ' + @MANV + N' KHÔNG THUỘC PHÒNG ' + CAST(@mapb AS
        nvarchar(3))
```

```
END
```

-- THỰC THI PROC

```
EXEC usp_thongkeNVtheoPB 123, 5
```

```
EXEC usp_thongkeNVtheoPB 123, 3
```

III. BÀI TẬP

2. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
 - a. Thêm vào một nhân viên cho bảng nhân viên. Nếu mã nhân viên được thêm vào đã có tồn tại thì in thông báo ‘không thêm được vì mã nhân viên đã tồn tại’, ngược lại thực hiện thao tác thêm nhân viên.

III. BÀI TẬP

2. Sử dụng CSDL QLDA tạo các Stored procedure thực hiện các yêu cầu sau:
- b. Thêm vào một phòng ban cho bảng phòng ban. Nếu mã phòng ban được thêm vào đã có tồn tại thì in thông báo ‘không thêm được vì mã phòng ban đã tồn tại’, ngược lại thực hiện thao tác thêm phòng ban.

--TẠO SP THÊM PHÒNG BAN

```
CREATE PROC usp_themphong @maphong int, @tenphong char(10), @trphg char(3), @ngaynhanchuc date  
AS
```

```
BEGIN  
    IF NOT EXISTS (SELECT * FROM PHONGBAN WHERE PHONGBAN.maphg = @maphong)  
        INSERT INTO PHONGBAN(maphg,tenphg,trphg,ngay_nhanchuc)  
        VALUES(@maphong,@tenphong,@trphg,@ngaynhanchuc)  
    ELSE  
        PRINT N'MÃ PHÒNG BAN ĐÃ TỒN TẠI'
```

```
END
```

-- THỰC THI SP

```
EXEC usp_themphong 7, N'Tổ chức', N'123', '12/20/2000'
```


III. BÀI TẬP

3. Sử dụng CSDL QLHD tạo các Stored procedure thực hiện các yêu cầu sau:

- a. Xuất thông tin hóa đơn của nhân viên khi nhập mã nhân viên.
- b. Xuất thông tin chi tiết các hóa đơn của một khách hàng khi nhập mã khách hàng
- c. Xuất thông tin các hóa đơn được lập trong 1 khoảng thời gian do người dùng nhập.
- d. Tìm hóa đơn của một khách hàng theo số điện thoại hoặc tên hoặc địa chỉ.