
Chương 5

Thủ tục – Store Procedure

Nội dung trình bày

1. Giới thiệu Store Procedure
2. Tạo Store Procedure
3. Sửa, xóa Store Procedure

1. Giới thiệu Store Procedure

1. Giới thiệu Store Procedure

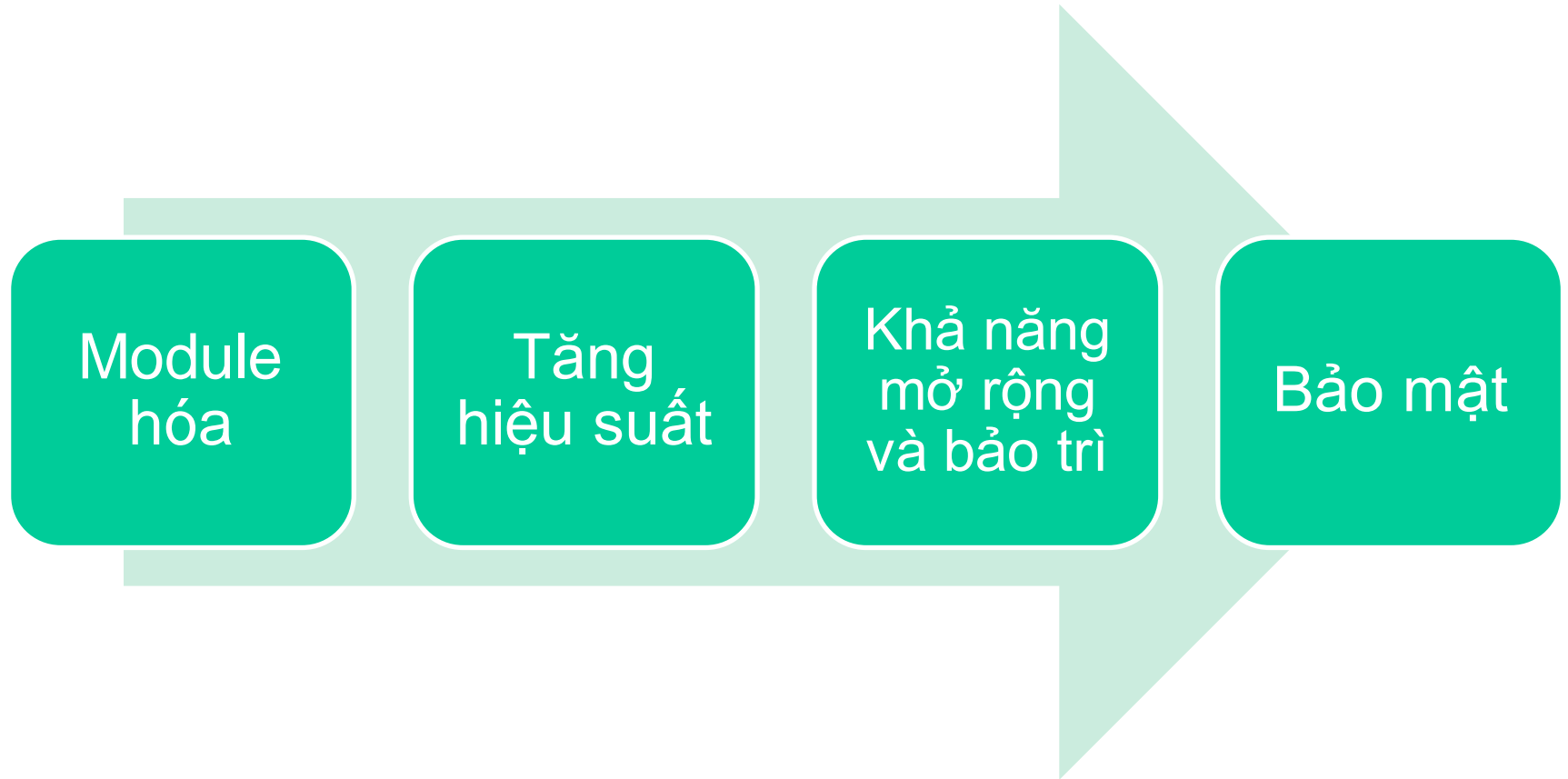
1.1. Giới thiệu: Stored Procedure (Thủ tục lưu trữ) là 1 phần không thể thiếu của SQL Server, **hỗ trợ rất nhiều cho lập trình và cấu hình cơ sở dữ liệu**

Stored Procedure là bao **gồm các câu lệnh Transact-SQL và được lưu lại trong cơ sở dữ liệu**

Store Procedure có **thể gọi và thực thi thông qua SQL Server Management Studio hoặc ngay trong ứng dụng đang phát triển**

1. Giới thiệu Store Procedure

1.2. Ưu điểm của Store Procedure:



1. Giới thiệu Store Procedure

Module hóa

Chỉ cần **viết Stored Procedure 1 lần**,
sau đó có thể **gọi nhiều lần ở trong
ứng dụng**

1. Giới thiệu Store Procedure

Tăng hiệu suất

Thực thi nhanh hơn: Stored Procedure sẽ **được biên dịch 1 lần và lưu ở dạng thực thi (tự động được lưu trữ)**. Sau đó có thể gọi thực thi nhiều lần mà không cần biên dịch lại

Giảm tải băng thông: Thay vì gửi lệnh SQL nhiều lần thì nên gom các câu lệnh SQL vào 1 Stored Procedure và **chỉ phải gọi đến 1 lần duy nhất qua network**.

1. Giới thiệu Store Procedure

Khả năng mở rộng và bảo trì

Nếu như ứng dụng có thay đổi j đó về mặt xử lý, có thể xem xét thay đổi lệnh ở Store Procedure và thực thi ở Sql Server. Khi đó không cần phải thay đổi mã lệnh của ứng dụng

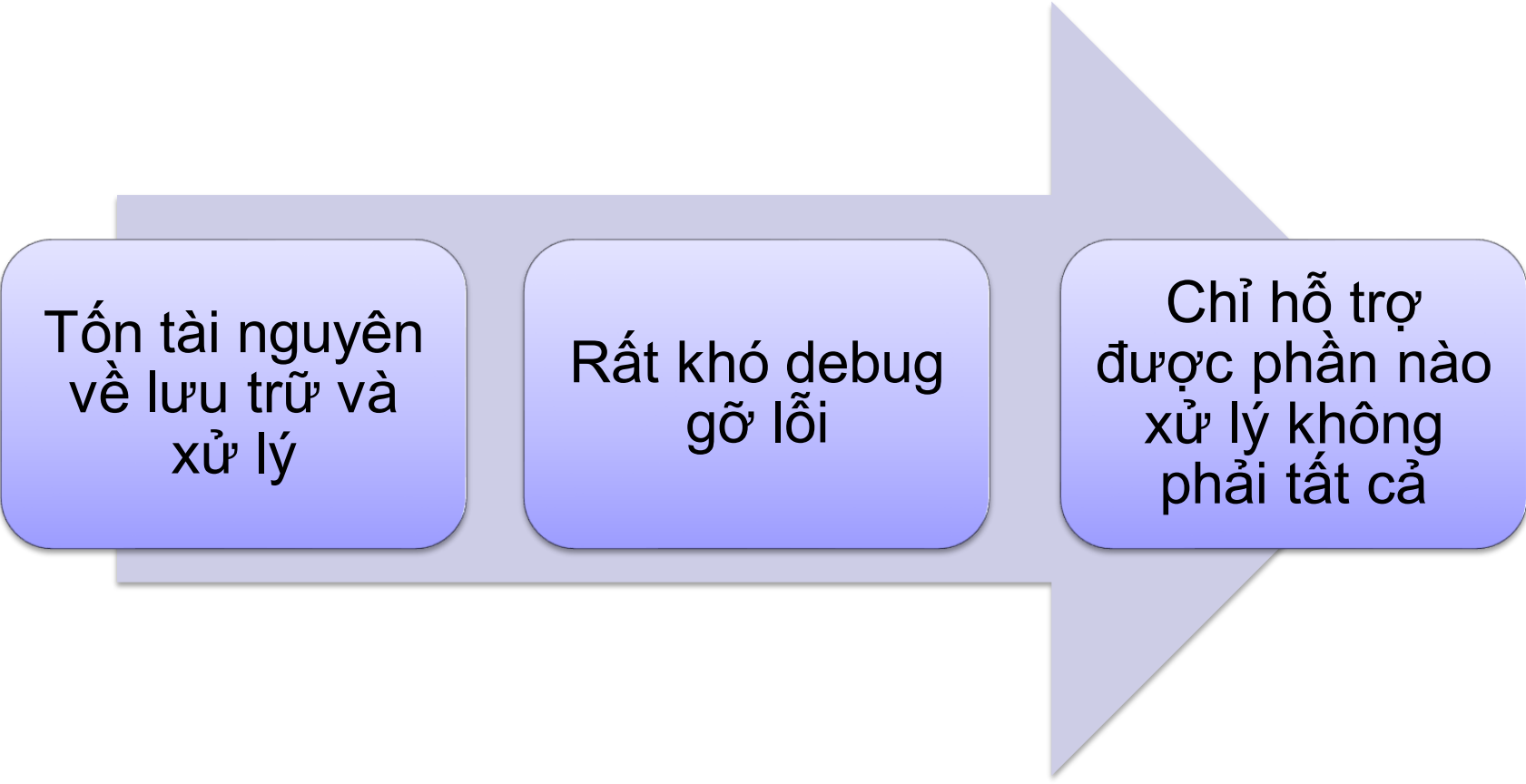
1. Giới thiệu Store Procedure

Bảo mật

Giới hạn việc truy xuất dữ liệu trực tiếp của một user nào đó vào một số tables không mong muốn

1. Giới thiệu Store Procedure

1.3. Nhược điểm của Store Procedure:



Tốn tài nguyên
về lưu trữ và
xử lý

Rất khó debug
gỡ lỗi

Chỉ hỗ trợ
được phần nào
xử lý không
phải tất cả

1. Giới thiệu Store Procedure

Tồn tài nguyên về
lưu trữ và xử lý

Thay vì tập trung vào mục đích chính là lưu trữ, thu nhận dữ liệu. Thì Sql Server phải làm thêm công việc xử lý tính toán nữa

1. Giới thiệu Store Procedure

Rất khó Debug

Rất khó để Debug Store Procedure trong Hệ QT CSDL, thường thì chỉ dựa vào kỹ năng của lập trình viên tìm Bug

1. Giới thiệu Store Procedure

Chỉ hỗ trợ được
phần nào xử lý
không phải tất cả

Bản chất của Sql Server vẫn là lưu trữ dữ liệu, do vậy Store Procedure chỉ hỗ trợ 1 phần thao tác xử lý, vẫn cần sử dụng đến các ngôn ngữ lập trình (C#, Java,..) kết hợp để ứng dụng được hoàn thiện

2. Tạo Store Procedure

2. Tạo Store Procedure

2.1. Tạo Store Procedure: Thủ tục lưu trữ được tạo bởi câu lệnh **CREATE PROCEDURE** với cú pháp như sau:

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

Tên_thủ_tục: Tên của thủ tục, tuân thủ quy tắc định danh và không vượt quá 128 ký tự

Danh_sách_tham_số: Tham số đầu vào của thủ tục, khai báo ngay sau tên thủ tục. Nếu có nhiều tham số đầu vào thì phải ngăn cách bằng dấu phẩy.

Cú pháp: <Tên tham số bắt đầu bằng dấu @> <Kiểu dữ liệu>

Ví dụ: @mamonhoc varchar(10)

RECOMPILE: Mặc định thủ tục sẽ được phân tích, tối ưu và dịch sẵn ở lần gọi đầu tiên, nếu sử dụng tùy chọn **WITH RECOMPILE** thì mỗi lần gọi thủ tục sẽ được biên dịch lại

ENCRYPTION: Thủ tục sẽ bị mã hóa nếu dùng tùy chọn **WITH ENCRYPTION**, khi đó 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. Để cấu trúc rõ ràng có thể đặt các câu lệnh trong **BEGIN END** hoặc có thể không

2. Tạo Store Procedure

Ví dụ: Tạo thủ tục hiển thị danh sách NhạcTruong

NhacTruong			
	Column Name	Condensed Type	Nullable
🔑	MaNhacTruo...	int	No
	TenNhacTruo...	nvarchar(50)	Yes

```
CREATE PROCEDURE SP_NhacTruong_GetList AS
BEGIN
    Select * from NhacTruong;
END
```


2. Tạo Store Procedure

Ví dụ: Tạo thủ tục thêm thông tin nhạc trường mới vào bảng NhạcTruong

NhacTruong			
	Column Name	Condensed Type	Nullable
🔑	MaNhacTruo...	int	No
	TenNhacTruo...	nvarchar(50)	Yes

```
CREATE PROCEDURE SP_NhacTruong_Insert
@MaNhacTruong int,
@TenNhacTruong nvarchar(50)
AS
BEGIN
insert into NhacTruong(MaNhacTruong, TenNhacTruong)
values(@MaNhacTruong, @TenNhacTruong);
END
```

2. Tạo Store Procedure

Ví dụ: Tạo thủ tục cập nhật tên nhạc trưởng

NhacTruong			
	Column Name	Condensed Type	Nullable
🔑	MaNhacTruo...	int	No
	TenNhacTruo...	nvarchar(50)	Yes

```
CREATE PROCEDURE SP_NhacTruong_Update
@MaNhacTruong int,
@TenNhacTruong nvarchar(50)
AS
BEGIN
    update NhacTruong
    set TenNhacTruong = @TenNhacTruong
    where MaNhacTruong= @MaNhacTruong;
END
```

2. Tạo Store Procedure

Ví dụ: Tạo thủ tục xóa nhạc trường

NhacTruong			
	Column Name	Condensed Type	Nullable
🔑	MaNhacTruo...	int	No
	TenNhacTruo...	nvarchar(50)	Yes

```
CREATE PROCEDURE SP_NhacTruong_Delete
@MaNhacTruong int
AS
BEGIN
    Delete from NhacTruong
    where MaNhacTruong= @MaNhacTruong;
END
```

2. Tạo Store Procedure

2.2. Gọi Store Procedure: Để gọi 1 thủ tục lưu trữ thực thi ta dùng cú pháp sau:

```
tên_thủ_tục [danh_sách_các_đối_số]
```

danh_sách_các_đối_số: Danh sách các tham số đầu vào truyền vào thủ tục, danh sách này phải tương ứng với số lượng và thứ tự khi tạo thủ tục lưu trữ

Trường hợp thủ tục lưu trữ được gọi trong một thủ tục khác, hoặc trigger hay kết hợp với câu lệnh SQL thì phải dùng từ khóa **EXECUTE**

```
EXECUTE tên_thủ_tục [danh_sách_các_đối_số]
```

2. Tạo Store Procedure

Ví dụ: Gọi thủ tục hiển thị danh sách nhạc trưởng

SP_NhacTruong_GetList

Hoặc

EXECUTE SP_NhacTruong_GetList

Ví dụ: Gọi thủ tục thêm 1 nhạc trưởng mới vào bảng nhạc trưởng

SP_NhacTruong_Insert 5, N'Trịnh Nam Sơn'

Hoặc

EXECUTE SP_NhacTruong_Insert 5, N'Trịnh Nam Sơn'

Ví dụ: Gọi thủ tục xóa 1 nhạc trưởng khỏi bảng nhạc trưởng

SP_NhacTruong_Delete 5

Hoặc

EXECUTE SP_NhacTruong_Delete 5

2. Tạo Store Procedure

Ví dụ: Gọi thủ tục thêm 1 nhạc trưởng mới vào bảng nhạc trưởng

```
SP_NhacTruong_Insert @MaNhacTruong = 5,  
@TenNhacTruong = N'Trịnh Nam Sơn'
```

Ví dụ: Gọi thủ tục xóa 1 nhạc trưởng khỏi bảng nhạc trưởng

```
SP_NhacTruong_Delete @MaNhacTruong = 5
```

2. Tạo Store Procedure

2.3. Khai báo biến trong Store 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
```

Tên biến **phải bắt đầu bởi ký tự @ và tuân theo qui tắc về định danh.**

2. Tạo Store Procedure

Ví dụ: Tạo thủ tục thêm thông tin nhạc trường mới vào bảng NhạcTruong

```
CREATE PROCEDURE SP_NhacTruong_Insert
@MaNhaTruong int, @TenNhaTruong nvarchar(50)
AS
BEGIN
    insert into NhaTruong(MaNhaTruong, TenNhaTruong)
        values(@MaNhaTruong, @TenNhaTruong);
END
```

→ Trong các bảng dữ liệu thì **mã số thường sử dụng làm khóa chính và sẽ tự động sinh ra khi thêm mới 1 dòng dữ liệu**. Với thủ tục lưu trữ trên có thể viết lại như sau:

```
CREATE PROCEDURE SP_NhacTruong_Insert
@TenNhaTruong nvarchar(50)
AS
BEGIN
    Declare @MaNhaTruongMoi int
    select @MaNhaTruongMoi = max(MaNhaTruong)+1 from NhaTruong;
    insert into NhaTruong(MaNhaTruong, TenNhaTruong)
        values(@MaNhaTruongMoi, @TenNhaTruong);
END
```


2. Tạo Store Procedure

Khi gọi thủ tục lưu trữ thêm 1 nhạc trưởng mới, chỉ cần truyền tên nhạc trưởng mới vào

SP_NhacTruong_Insert N'Hoàng Hiệp'

2. Tạo Store Procedure

2.4. Giá trị trả về của tham số trong Store Procedure

Có thể lấy giá trị đã thay đổi trong thủ tục lưu trữ ra ngoài bằng cách:

Khi tạo thủ tục lưu trữ: khai báo từ khóa **OUTPUT** (hoặc **OUT**) phía sau kiểu dữ liệu của tham số đầu vào

Khi gọi thủ tục lưu trữ: Đặt từ khóa **OUTPUT** (hoặc **OUT**) sau tham số truyền vào

2. Tạo Store Procedure

```
CREATE PROCEDURE sp_CongHaiSo @a INT, @b INT, @c INT)
AS
BEGIN
    SELECT @c=@a+@b
END
```

```
DECLARE @tong INT
```

```
SELECT @tong=0
```

```
EXECUTE sp_CongHaiSo 100, 200, @tong
```

```
SELECT @tong
```

Results		Messages	
KETQUA			
1	0		

2. Tạo Store Procedure

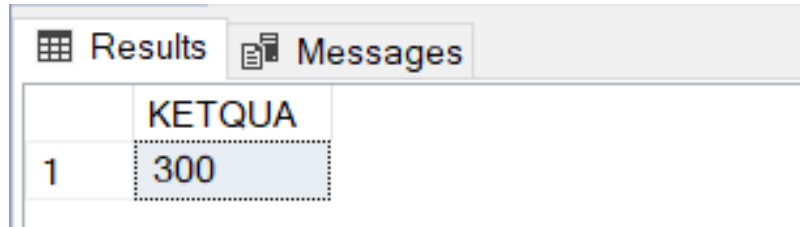
```
CREATE PROCEDURE sp_CongHaiSo @a INT, @b INT, @c INT OUTPUT)
AS
BEGIN
    SELECT @c=@a+@b
END
```

```
DECLARE @tong INT
```

```
SELECT @tong=0
```

```
EXECUTE sp_CongHaiSo 100, 200, @tong OUTPUT
```

```
SELECT @tong
```



The screenshot shows a SQL Server interface with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with one column named 'KETQUA' and one row containing the value '300'.

	KETQUA
1	300

2. Tạo Store Procedure

2.5. Tham số với giá trị 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

```
@tên_tham_số kiểu_dữ_liệu = giá_trị_mặc_định
```

2. Tạo Store Procedure

2.5. Tham số với giá trị 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

```
@tên_tham_số kiểu_dữ_liệu = giá_trị_mặc_định
```

2. Tạo Store Procedure

Ví dụ: Tạo thủ tục thêm thông tin nhạc trường mới vào bảng NhạcTruong, Nếu không truyền tên nhạc trường thì tên nhạc trường mang giá trị 'NULL'

```
CREATE PROCEDURE SP_NhacTruong_Insert
@TenNhacTruong nvarchar(50) = 'NULL'
AS
BEGIN
    Declare @MaNhacTruongMoi int
    select @MaNhacTruongMoi = max(MaNhacTruong)+1
    from NhacTruong;
    insert into NhacTruong(MaNhacTruong, TenNhacTruong)
        values(@MaNhacTruongMoi, @TenNhacTruong);
END
```

2. Tạo Store Procedure

Gọi thủ tục thêm mã nhạc trưởng mới

Gọi thủ tục không truyền tham số, 1 dòng mới sẽ được thêm vào bảng nhạc trưởng với tên là 'NULL'

SP_NhacTruong_Insert

Gọi thủ tục có truyền tham số, 1 dòng mới được thêm vào bảng nhạc trưởng với tên 'Nguyễn Văn A'

SP_NhacTruong_Insert N'Nguyễn Văn A'

3. Sửa, Xóa Store Procedure

3. Sửa, xóa Store Procedure

3.1. Sửa Store Procedure: Khi một thủ tục đã được tạo ra, có thể tiến hành định nghĩa lại thủ tục đó bằng câu lệnh

ALTER PROCEDURE có cú pháp như sau

```
ALTER PROCEDURE tên_thủ_tục [(danh_sách_tham_số)]  
[WITH RECOMPILE|ENCRYPTION|RECOMPILE,ENCRYPTION]  
AS
```

Các_câu_lệnh_Của_thủ_tục

3. Sửa, xóa Store Procedure

```
CREATE PROCEDURE SP_NhacTruong_Insert
```

```
@TenNhacTruong nvarchar(50) = 'NULL' AS
```

```
BEGIN
```

```
    Declare @MaNhacTruongMoi int
```

```
    select @MaNhacTruongMoi = max(MaNhacTruong)+1
```

```
    from NhacTruong;
```

```
    insert into NhacTruong(MaNhacTruong, TenNhacTruong)
```

```
        values(@MaNhacTruongMoi, @TenNhacTruong);
```

```
END
```

```
ALTER PROCEDURE SP_NhacTruong_Insert
```

```
@TenNhacTruong nvarchar(50) AS
```

```
BEGIN
```

```
    Declare @MaNhacTruongMoi int
```

```
    select @MaNhacTruongMoi = max(MaNhacTruong)+1
```

```
    from NhacTruong;
```

```
    insert into NhacTruong(MaNhacTruong, TenNhacTruong)
```

```
        values(@MaNhacTruongMoi, @TenNhacTruong);
```

```
END
```

3. Sửa, xóa Store Procedure

3.2. Xóa Store Procedure: Để xóa một thủ tục đã có, ta sử dụng câu lệnh **DROP PROCEDURE** với cú pháp như sau:

DROP PROCEDURE tên_thủ_tục

Khi xóa một thủ tục, tất cả các quyền đã cấp cho người sử dụng trên thủ tục đó cũng đồng thời bị xóa bỏ. Do đó, nếu tạo lại thủ tục, ta phải tiến hành cấp phát lại các quyền trên thủ tục đó.

Ví dụ: Xóa thủ tục **SP_NhacTruong_Insert**

Drop Procedure SP_NhacTruong_Insert

Câu hỏi ???