Stored Procedure

**Stored Procedure (SP) là**gì**?**

**Stored Procedure (Thủ tục lưu trữ)**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:

* Các cấu trúc điều khiển (IF, WHILE, FOR) có thể được sử dụng trong thủ tục.
* Bên trong thủ tục lưu trữ có thể sử dụng các biến như trong ngôn ngữ lập trình nhằm lưu giữ các giá trị tính toán được, các giá trị được truy xuất được từ cơ sở dữ liệu.
* Một tập các câu lệnh SQL được kết hợp lại với nhau thành một khối lệnh bên trong một thủ tục. Một thủ tục có thể nhận các tham số truyền vào cũng như có thể trả về các giá trị thông qua các tham số (như trong các ngôn ngữ lập trình). Khi một thủ tục lưu trữ đã được định nghĩa, nó có thể được gọi thông qua tên thủ tục, nhận các tham số truyền vào, thực thi các câu lệnh SQL bên trong thủ tục và có thể trả về các giá trị sau khi thực hiện xong.

**Lợi ích khi sử dụng SP**

* Khả năng module hoá: Bạn có thể tạo một SP một lần và tái sử dụng nó bao nhiêu lần tùy thích trong chương trình của bạn. Điều này làm tăng khả năng bảo trì ứng dụng của bạn và cho phép các ứng dụng truy cập vào các cơ sở dữ liệu trong một cách thức thống nhất và tối ưu hóa.
* Đơn giản hoá các thao tác trên cơ sở dữ liệu nhờ vào khả năng module hoá các thao tác này.
* Thủ tục lưu trữ được phân tích, tối ưu khi tạo ra nên việc thực thi chúng nhanh hơn nhiều so với việc phải thực hiện một tập rời rạc các câu lệnh SQL tương đương theo cách thông thường.
* Thủ tục lưu trữ cho phép chúng ta thực hiện cùng một yêu cầu bằng một câu lệnh đơn giản thay vì phải sử dụng nhiều dòng lệnh SQL. Điều này sẽ làm giảm thiểu sự lưu thông trên mạng.
* Thay vì cấp phát quyền trực tiếp cho người sử dụng trên các câu lệnh SQL và trên các đối tượng cơ sở dữ liệu, ta có thể cấp phát quyền cho người sử dụng thông qua các thủ tục lưu trữ, nhờ đó tăng khả năng bảo mật đối với hệ thống.
* SP có thể được gọi từ các chương trình phía client hoặc từ các SP khác do đó SP được ưa chuộng trong lập trình CSDL, đặc biệt là trong các ứng dụng theo mô hình N-tier và các Web services.

**Tạo thủ tục lưu trữ**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

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.  
*– 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  
*–  RECOMPILE:* Thông thường, thủ tục sẽ được phân tích, tối ưu và dịch sẵn ở lần gọi đầu tiên. Nếu tuỳ chọn WITH RECOMPILE được chỉ định, thủ tục sẽ được dịch lại mỗi khi được gọi.  
*– ENCRYPTION:* Thủ tục sẽ được mã hoá nếu tuỳ 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. Các câu lệnh này có thể đặt trong cặp từ khoá BEGIN…END hoặc có thể không + Để thực thi một thủ tục ta thực hiện: Execute Ten\_Store\_Procedure Danh\_sách\_các\_tham\_số (Hoặc EXEC  Ten\_Store\_Procedure Danh\_sách\_các\_tham\_số)  
+ Để chỉnh sửa một Stored Procedure có sẵn: sử dụng lệnh Alter Procedure  
+ Để xóa một Stored Procedure: sử dụng lệnh Drop Procedure  Ten\_Store\_Procedure  
+ Để đổi tên một Stored Procedure ta dùng lệnh: sp\_rename ‘tên\_sp\_cũ’ ‘tên\_sp\_mới’  
+ Để xem nội dung Stored Procedure ta dùng sp\_helptext tên\_sp

**Ví dụ 1:** Tạo Store Procedure không có tham số  
Trong định nghĩa chúng ta có [(*Danh\_sách\_tham\_số*)] Nhưng đôi khi Store có thể không có Danh sách các tham số. Bạn tham khảo ví dụ sau

CREATE PROCEDURE spCategories\_GetCate\_ViDu1

AS

BEGIN

SELECT c.CategoryID,

c.CategoryName,

c.Description

FROM Categories c

END

Trong ví dụ trên chúng ta đã tạo một SP là spCategories\_GetCate để truy vấn các trường CategoryID, CategoryName, Description của bảng Categories, Bạn có thể chú ý đến cách đặt tên tuy tên của SP là tùy chọn nhưng để cho dễ kiểm soát khi Database của bạn lớn Bạn nên đặt một quy tắc đặt tên cho SP. Chúng ta thường đặt là spTênBảng\_ChứcnăngcủaSP vì khi database của bạn có thể sẽ có rất nhiều bảng và rất nhiều store nêu không theo quy tắc khi cần Alter một SP bạn sẽ khó tìm SP đó. (Trong SQL server có hỗ trợ chức năng Filter – Phải chuột vào Store Procedure/Filter/Filter setting sau đó nhập từ khóa cần Filter)

**Ví dụ 2:** Tạo Store Procedure  có tham số đầu vào.  
Trong ví dụ sau ta sẽ tạo 1 SP có tham số đầu vào và truy vấn theo điều kiện của tham số

CREATE PROCEDURE spCategories\_GetCate\_ViDu2

@CategoryID int

AS

BEGIN

IF @CategoryID>0

SELECT c.CategoryID,

c.CategoryName,

c.Description

FROM Categories c

WHERE c.CategoryID=@CategoryID

ELSE

SELECT c.CategoryID,

c.CategoryName,

c.Description

FROM Categories c

ORDER BY c.CategoryID DESC

END

Trong ví dụ trên bạn thấy có tham số đầu vào là @CategoryID khi @CategoryID =0 thì sẽ truy vấn All Còn khi @CategoryID >0 thì truy vấn theo @CategoryID  
Đây cũng thêm một cách mà chúng ta hay dùng để giảm bớt số lượng SP trong database. (Thường thì bạn sẽ phải viết 2 SP một là Select One – @CategoryID >0, 2 là Select All @CategoryID =0). Bạn thấy rằng trong SP trên chúng ta có sử dụng cấu trúc điều khiển IF. Bạn có thể thấy là SQL cũng có thể lập trình và có thể nó sẽ mang lại nhiều lợi ích khi lập trình trên SQL. Tham khảo thêm bài viết sau: [**Hướng dẫn xây dựng website**](http://hmweb.com.vn/157/pro/71/61/cntt.aspx)

**Ví dụ 3:** Store Procedure có tham số đầu vào – thực hiện Insert, Delete, Update.  
Trong Store Procedure sau chúng ta sẽ minh họa việc tạo một SP để thực hiện Thêm, Sửa, Xóa một bản ghi phụ thuộc vào điều kiện của tham số:

CREATE PROCEDURE spCategories\_Edit\_ViDu3

@Action int,

@CategoryID int,

@CategoryName nvarchar(50),

@Description nvarchar(400),

@Picture image

AS

BEGIN

IF @Action=0

DELETE FROM Categories

WHERE CategoryID=@CategoryID

ELSE

BEGIN

IF @CategoryID=0

INSERT INTO Categories(

[CategoryName],

[Description],

[Picture])

VALUES (

@CategoryName,

@Description,

@Picture

)

ELSE

UPDATE Categories

SET

CategoryName = @CategoryName,

Description = @Description,

Picture = @Picture

WHERE CategoryID=@CategoryID

END

END

 Trong ví dụ trên bạn thấy chúng ta sử dụng 1 Store để thực hiện 3 hành động thêm, sửa, xóa một bản ghi trong bảng Categorys. Cũng là một cách chúng ta muốn giới thiệu để giảm bớt số lượng SP trong cơ sở dữ liệu của bạn.**Ví dụ 4:**Store Procedure có tham số đầu vào, Tham số đầu ra và trả về giá trị

CREATE PROCEDURE spCategories\_Edit\_ViDu4

@Action int,

@CategoryID int,

@CategoryName nvarchar(50),

@Description nvarchar(400),

@Picture image,

@error nvarchar(1000) output

AS

BEGIN TRAN

IF @Action=0

BEGIN

DELETE FROM Categories

WHERE CategoryID=@CategoryID

IF @@ERROR<>0

BEGIN

SET @error = N'Lỗi: Chưa xóa được'

IF @@TRANCOUNT>0

ROLLBACK tran

GOTO Error

END

ELSE

SET @error = N'Đã xóa bản ghi'

END

ELSE

IF @CategoryID=0

BEGIN

INSERT INTO Categories(

[CategoryName],[Description],[Picture])

VALUES (

@CategoryName,@Description,@Picture

)

IF @@ERROR<>0

BEGIN

SET @error = N'Lỗi: Lỗi khi thêm dữ liệu'

IF @@TRANCOUNT>0

ROLLBACK tran

GOTO Error

END

ELSE

SET @error = N'Đã thêm mới bản ghi'

END

ELSE

BEGIN

UPDATE Categories

SET

[CategoryName] = @CategoryName,

[Description] = @Description,

[Picture] = @Picture

WHERE CategoryID=@CategoryID

IF @@ERROR<>0

BEGIN

SET @error = N'Lỗi: Lỗi khi cập nhật dữ liệu'

IF @@TRANCOUNT>0

ROLLBACK tran

GOTO Error

END

ELSE

SET @error = N'Đã cập nhật dữ liệu'

END

RETURN @error -- Select @error as ThongBao

COMMIT TRAN

Error:

Bạn để ý thấy rằng trong khi thực hiện SP trên thì ở trường hợp nào bạn cũng sẽ nhận được giá trị trả về tương ứng thông báo cho bạn biết trạng thái thực thi. Bạn cũng nên lưu ý sử dụng biên Output trong SP và dùng cách dùng ROLLBACK tran. Khi trong SP của bạn thực hiện nhiều hành động (Có thể là thực hiện Edit dữ liệu nhiều bảng khác nhau trong 1 SP khi đã thực hiện được 1 lệnh nào đó và đến lệnh tiếp theo xảy ra lỗi khi đó nếu  bạn không dùng dùng ROLLBACK thì sẽ xảy ra sai dữ liệu mà lại khó tím lỗi). Tham khảo thêm bài viết: [**Những thói quen tốt khi viết câu lệnh SQL**](http://hmweb.com.vn/239/pro/71/61/cntt.aspx). Trong những bài viết sau của loạt bài hướng dẫn này chúng ta sẽ giới thiệu cách dùng C# để thực thi một Store Procedure