7

Thủ tục thường trú

STORED PROCEDURES

Nguyễn Trong Anh
E-Mail: anh@tronganh.com
tronganh@gmail.com
Home: http://www.tronganh.com

Thủ tục thường trú

Là tập hợp các T-SQL được lưu trữ trong tên, được thực hiện như một đối tượng riêng biệt

Ưu điểm

Tăng tốc độ thực hiện

Giảm lưu lượng dao dịch trên mạng

Toàn vẹn dự liệu tăng

Bảo mật tốt hơn

Dạng của Thủ tục thường trú

- Người dùng định nghĩa (user – define)
- Hệ thống (system)
- Tạm thời (temporary)
- Tách biệt (remote)
- Mở rộng (Extended)

Thủ tục thường trú
Stored Procedure

- Khái niệm cơ bản về thủ tục thường trú
- Thay đổi và xoá một thủ tục thường trú
- Tham số và khai báo biến
- Phát biểu có cấu trúc
- Một số thủ tục thường trú cơ bản
- Một số thủ tục thường trú của hệ thống

Khái niệm

Thủ tục thường trú là một đối tượng xây dựng bởi những phát biểu của SQL server và T-SQL

Thủ tục thường trú được lưu trữ như một phần của cơ sở dữ liệu.

Cấu trúc như là văn bản Text, mỗi khi thực hiện chỉ cần gọi tương tự như thủ tục hoặc hàm trong các ngôn ngữ lập trình

Cú pháp để tạo thủ tục thường trú

```
CREATE PROCEDURE ten_thutuc
[<Cac tham so>],
[<Cac gia tri mac dinh>]
AS
BEGIN
        caulenh_sql1
        caulenh_sql2
 END
```

Xác định các thông tin cần thiết để tạo thủ tục thường trú

Nơi tạo thủ tục thường trú: Cơ sở dữ liệu NorthWind

Kiểu của thủ tục thường trú: user-defined

Tên của thủ tục thường trú: sp_Hienthi

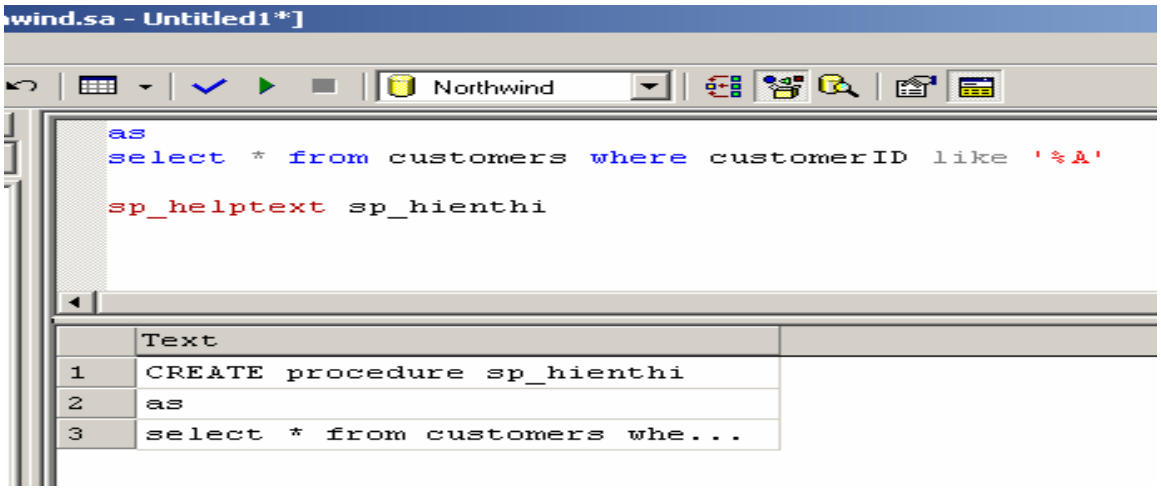
Nội dung của thủ tục sp_hienthi

```
Create procedure sp_hienthi
As
Begin
    select * from customers where customerID like
'%A‘
End
```

Xem nội dung của thủ tục thường trú

Sp_helptext sp_hienthi

Nội dung của thủ tục sẽ được hiển thị như hình vẽ dưới đây:



Gọi thủ tục thường trú

Câu lệnh EXECUTE PROCEDURE thường được dùng để gọi thủ tục thường trú

Cú pháp:

EXECUTE ten_thutuc

hoặc

EXEC ten_thutuc

hoặc

ten_thutuc

Tham số trong thủ tục thường trú

Khi thực hiện một thủ tục thường trú, chúng ta có thể truyền tham số để thông báo cho thủ tục thường trú

Có hai loại tham số

- Input parameter
- Output parameter

Tham số trong thủ tục thường trú

Khi sử dụng tham số trong thủ tục thường trú, chúng ta phải qua. Để khai báo một tham số, chúng ta cần quan tâm đến các yếu tố sau:

- Tên tham số
- Kiểu dữ liệu
- Giá trị mặc nhiên nếu có
- Có hay không chỉ dẫn OUTPUT

Tham số trong thủ tục thường trú (Khai báo tham số)

Cú pháp:

@ten_thamso [AS] kiểudữliệu

Ví dụ khai báo tham số

@SoDienThoai varchar(20),

@Diachi AS varchar(50)

Khi có nhiều tham số chúng ta sử dụng dấu (,) để phân cách các tham số

Tham số trong thủ tục thường trú (Khai báo tham số trong thủ tục)

```
Create procedure sp_hienthiTS
@Val Varchar(20)
as
Begin
 select * from customers where customerID like
'%' + @Val
End
```

Tham số trong thủ tục thường trú (Gọi thủ tục với tham số)

Xem nội dung thủ tục

Sp_helptext sp_hienthi

Gọi thủ tục

Sp_thutuc ‘A’

Ý nghĩa: Hiển thị thông tin của khách hàng. Với điều kiện CustomerID có ký tự A

Tham số trong thủ tục thường trú (Khai báo tham số trong thủ tục)

Những tham số được truyền từ bên ngoài vào thủ tục. Các tham số có thể lấy giá trị truyền vào tư ngôn ngữ lập trình. Ví dụ như: Visual Basic, Visual Basic.Net, C# …

Giá trị truyền vào phải đúng thứ tự như đã khai báo. Nếu giá trị truyền vào không tương thích với kiểu dữ liệu đã được khai báo. Lỗi sảy ra ➔ Không chạy được thủ tục

Xây dựng thủ tục với các giá trị mặc định

Khi khai báo tham số trong thủ tục thường trú, nếu cần chúng ta có thể khởi tạo giá trị mặc nhiên cho tham số.

Khi gọi thủ tục có gán giá trị ngẫu nhiên, nếu người sử dụng không cung cấp giá trị. Nó sẽ lấy giá trị mặc định được định nghĩa trước đó.

Tạo thủ thục tham số với giá trị mặc định

```
CREATE procedure sp_InsertShipper
@company Varchar(20) = 'N/A',
@Phone Varchar(20) = 'N/A'
AS
Begin
 Insert Into Shippers(CompanyName,Phone)
 Values(@company, @phone)
End
GO
```

Tạo thủ thục tham số với giá trị mặc định

Thủ tục trên thêm mới 1 bản ghi vào bảng Shippers

Nếu có giá trị truyền vào. Thủ tục sẽ lấy các giá trị được truyền vào.

Giá trị truyền vào khi gọi tham số cho companyName và Phone sẽ được lấy mặc định nếu là rỗng

Kiểm tra kết quả của thủ tục

Chúng ta chạy Thủ tục trên với các tham số dưới đây:

sp_InsertShipper

sp_InsertShipper ‘Dai Học Thang Long’

sp_InsertShipper ‘DH Thang Long’, ‘858 789 989’

Thay đổi thủ tục thường trú

Những lưu ý khi thay đổi nội dung của Thủ tục thường trú

- Thủ tục đó phải tồn tại.

- Tuỳ thuộc vào quyền hạn của người dùng đó có thể thay đổi thủ tục thường trú hay không.

- Kiểm tra tất cả các thông tin có liên quan đến các đối tượng khác trong khi bị thay đổi.

Thủ tục thường trú với tham số (*Tham số Output*)

Lấy giá trị Output:

- Khi chúng ta cần xuất giá trị ra ngoài khi thủ tục thường trú thực thi xong.

- Sử dụng kết quả của thủ tục thường trú làm giá giá trị tham số đầu vào cho một thủ tục thường trú khác.

Thủ tục thường trú với tham số (*Tham số Output*)

```
CREATE procedure sp_InsertShipper1
    @company Varchar(20) = 'N/A',
    @Phone Varchar(20) = 'N/A',
    @outPhone int OutPut
AS
Begin
   Insert Into Shippers(CompanyName,Phone)
   Values(@company, @phone)
   select @outPhone = @@IDentity
End
GO
```

Thủ tục thường trú với tham số (*Lấy thông tin từ Tham số Output*)

```
declare @MyID int
exec sp_InsertShipper1 'Test1','Test2', @MyID
OutPut

select @myID as SoMauTin

select * from shippers where ShipperID = @myID
```

Thủ tục thường trú với tham số (*Nhiều Tham số Output*)

```
CREATE PROCEDURE prcGetInfoShippers
        @ShipperID int,
        @Phone varchar(20) OUTPUT,
       @company varchar(20) OUTPUT
           AS
BEGIN
   SELECT
    @Phone = Phone,
    @company = CompanyName
    FROM Shippers
    WHERE ShipperID = @ShipperID
END
```

Thủ tục thường trú với tham số (*lấy thông tin từ thủ tục nhiều Tham số Output*)

Khai báo biến

Declare @Phone Varchar(20)

Declare @Company Varchar(20)

Gọi Thủ tục

Exec prcGetInfoShippers 1, @Phone Output, @Company Output

In kết quả

Select @Phone as Phone, @Company as Company

Những câu lệnh điều khiển

Những câu lệnh điều khiển

Bất kỳ ngôn ngữ lập trình nào hiện nay đều có những phát biểu điều khiển.

SQL Server 2000 cũng giống như ngôn ngữ lập trình.

Chúng ta tìm hiểu các cú pháp thông dụng như:

- IF … ELSE
- GOTO
- WHILE
- WAITFOR

Những câu lệnh điều khiển (2)

Bên cạnh đó còn có các câu lệnh điều khiển như:

- CASE …
- WHEN …
- THEN …
- ELSE … END

Câu lệnh IF ... ELSE

Câu lệnh IF … ELSE được sử dụng nhiều trong các ngôn ngữ lập trình.

Câu lệnh IF … ELSE đóng vai trò rất quan trọng khi lập trình, ngay cả trong Stored Procedure (thủ tục thường trú), hoặc trong trigger (Chúng ta sẽ tìm hiểu phần này sau)

Câu lệnh IF ... ELSE

Cú pháp:

IF <biểu thức logic>
    Begin <Các câu lệnh SQL> End
ELSE
    Begin <Các câu lệnh SQL> End

<Các câu lệnh SQL>: có thể là 1 hoặc nhiều câu lệnh SQL

Câu lệnh IF ... ELSE (Minh hoạ)

```
Create PROCEDURE prcGetInfoShippers32
          @ShipperID int,
        @Phone char(20) OUTPUT
           AS
BEGIN
    IF EXISTS(SELECT * FROM Shippers WHERE ShipperID =
           @ShipperID)
    BEGIN
     SELECT
        @Phone = Phone
           FROM Shippers
      WHERE ShipperID = @ShipperID
      END
    ELSE
     Select @Phone = 'Not Found'
END
```

Câu lệnh IF ... ELSE (Minh hoạ)

Xem kết quả gọi hàm (trường hợp có dữ liệu)

- Declare @Phone varchar(20)
- exec prcGetInfoShippers32 1,@phone OUTPUT
- Select @Phone as SoDienThoai

Xem kết quả gọi hàm (trường hợp không có dữ liệu)

- Declare @Phone varchar(20)
- exec prcGetInfoShippers32 0,@phone OUTPUT
- Select @Phone as SoDienThoai

Câu lệnh CASE

- Câu lệnh CASE cho phép nhận một giá trị từ nhiều lựa chọn.
- Trường hợp có nhiều câu lệnh IF ELSE lồng nhau, gây cho đoạn chương trình phức tạp, bạn nên sử dụng câu lệnh CASE.

- Để dễ hiểu câu lệnh CASE. Cú pháp Câu lệnh CASE giống như các ngôn ngữ lập trình khác như: C, C++, Java …

Cú pháp CASE

CASE <giá trị biểu thức Case>

- WHEN <điều kiện 1> THEN <kết quả tương ứng 1>
- WHEN <điều kiện 2> THEN <kết quả tương ứng 2>
- WHEN <điều kiện 3> THEN <kết quả tương ứng 3>
- …
- ELSE <kết quả tương ứng >

END

Câu lệnh Case
(*Ví dụ minh hoạ với Select*)

```
Create procedure sp_ConvertStatess
AS
Begin
 Select ShipperID, CompanyName, Phone,
 Case State
  WHEN 'CA' then  'Califorlia'
  WHEN 'KS' then  'Kansas'
  WHEN 'TN' Then 'Tennessee'
  WHEN 'OR' Then 'Oregon'
  WHEN 'MI' Then 'Michigan'
  WHEN 'IN' Then 'India'
  WHEN 'MD' Then 'Marylan'
  WHEN 'UT' Then 'Utah'
  ELSE 'Khong xac dinh'
 End
 From Shippers
End
```

Câu lệnh Case
(*Ví dụ minh hoạ với Select*)

Câu lệnh CASE với câu truy vấn SELECT ở trên sẽ có hai trường hợp sảy ra:

- Nếu Không có biểu thức ELSE, kết quả trả về chỉ tương ứng với các điều kiện của các biểu thức WHEN.

- Nếu có biểu thức ELSE, giá trị tương ứng của biểu thức ELSE sẽ được hiển thị khi không có các giá trị trong các biểu thức WHEN

Câu lệnh Case

(*Ví dụ minh hoạ với khi tính toán với biểu thức logic*)

```
Create procedure sp_ConvertStateAndPrice
AS
Begin
 Select ShipperID, CompanyName, Phone,
Case
 WHEN Price < 0.5 Then 0.5
 WHEN Price Between 1.00 and 2.00 Then 2.3
 WHEN Price Between 2.3 and 4 Then 4
 ELSE 5
 END
 From Shippers
End
```

Ví dụ với 2 CASE

```
Create procedure sp_ConvertStateAndPrice
AS
Begin
 Select ShipperID, CompanyName, Phone,
 Case State
  WHEN 'CA' then  'Califorlia'
  WHEN 'KS' then  'Kansas'
  WHEN 'TN' Then 'Tennessee'
  WHEN 'OR' Then 'Oregon'
  WHEN 'MI' Then 'Michigan'
  WHEN 'IN' Then 'India'
  WHEN 'MD' Then 'Marylan'
  WHEN 'UT' Then 'Utah'
  ELSE 'Khong xac dinh'
 End,
 Case
 WHEN Price < 0.5 Then 0.5
 WHEN Price Between 1.00 and 2.00 Then 2.3
 WHEN Price Between 2.3 and 4 Then 4
 ELSE 5
 END
 From Shippers
End
```

Câu lệnh While

- Câu lệnh While là câu lệnh điều khiển vòng lặp.
- Vòng lặp sẽ được thực hiện cho đến khi biểu thức trong While sai.
- While được dùng nhiều trong kiểu dữ liệu CURSOR. Thông thường While thường dùng để duyệt từ bản ghi đầu tiên đến bản ghi cuối cùng hoặc ngược lại

Cú pháp Câu lệnh While

While <biểu thức logic>
  <các câu lệnh SQL>
Begin
 <các khối lệnh bị cấm> (Block statements)
 <Breake>
 <Các câu lệnh SQL>
 [Continue]
End

Cú pháp Câu lệnh While

BREAKE dùng để thoát khỏi vòng lặp While. Ví dụ khi trong thủ tục thường trú. Nếu gặp Breake nó sẽ thoát khỏi vòng lặp các lệnh phía sau sẽ bị bỏ qua.

Continue ngược lại với BREAKE, nếu gặp câu lệnh này thì quá trình xử lý sẽ quay lại đầu vòng lặp While

Cú pháp Câu lệnh While (VD)

```
Create procedure sp_UpdateShipper
AS
Begin
 declare @i int
 set @i = 1
 While @i <= 5
 Begin
  update Shippers set price = price + 10 where shipperID =
@i
    set @i = @i+1
 End
End
```

Câu lệnh RETURN

- Khi cần xác định kết quả đúng hoặc trả về một giá trị nào đó. Chúng ta sử dụng câu kệnh RETUNS

- Nếu gặp câu lệnh RETURN, quá trình sử lý được kết thúc.

- Trong thủ tục đôi khi chúng ta sử dụng RETURN để thủ tục trở thành hàm như các ngôn ngữ lập trình khác,

Câu lệnh RETURN

Cú pháp:

Return <giá trị nguyên>

Nếu không cung cấp giá trị trả về cho câu lệnh Return, giá trị trả về sẽ là 0

Câu lệnh Return (minh hoạ) (giá trị trả về là giá trị chỉ định)

Tính tổng số lượng trong bảng Shippers

```
Create procedure sp_GetSumQuantity
AS
Begin
 declare @SumQuantity int
 Select @SumQuantity = Sum(Quantity) from
Shippers
 Return @SumQuantity
End
```

Tính số ngày của tháng (Return)

```
Create Procedure sp_days
@thangnam varchar(7)
AS
Begin
 Declare @nam int
 Declare @thang int
 Declare @songay int
 set @nam = cast(right(@thangnam,4) as int)
 set @thang = cast(left(@thangnam,2) as int)
 Set @songay =
 Case
   When @thang in (1,3,5,7,8,10,12) then 31
   When @thang in (4,6,9,11) Then 30
   When @thang in (2) and (@nam %4 = 0 and @nam%100 = 0)
then 29
   Else 28
 End
 Return @songay
End
```

Tính số ngày của tháng (Return)

Chạy thủ tục trên

```
Declare @days int

exec @days = sp_days '10/2001'

Print 'So ngay cua thang la: ' + str(@days)
```

Tính số ngày của tháng

Đối với thủ tục trên chúng ta đã sử dụng một số cú pháp mà chưa đề cập từ trước:

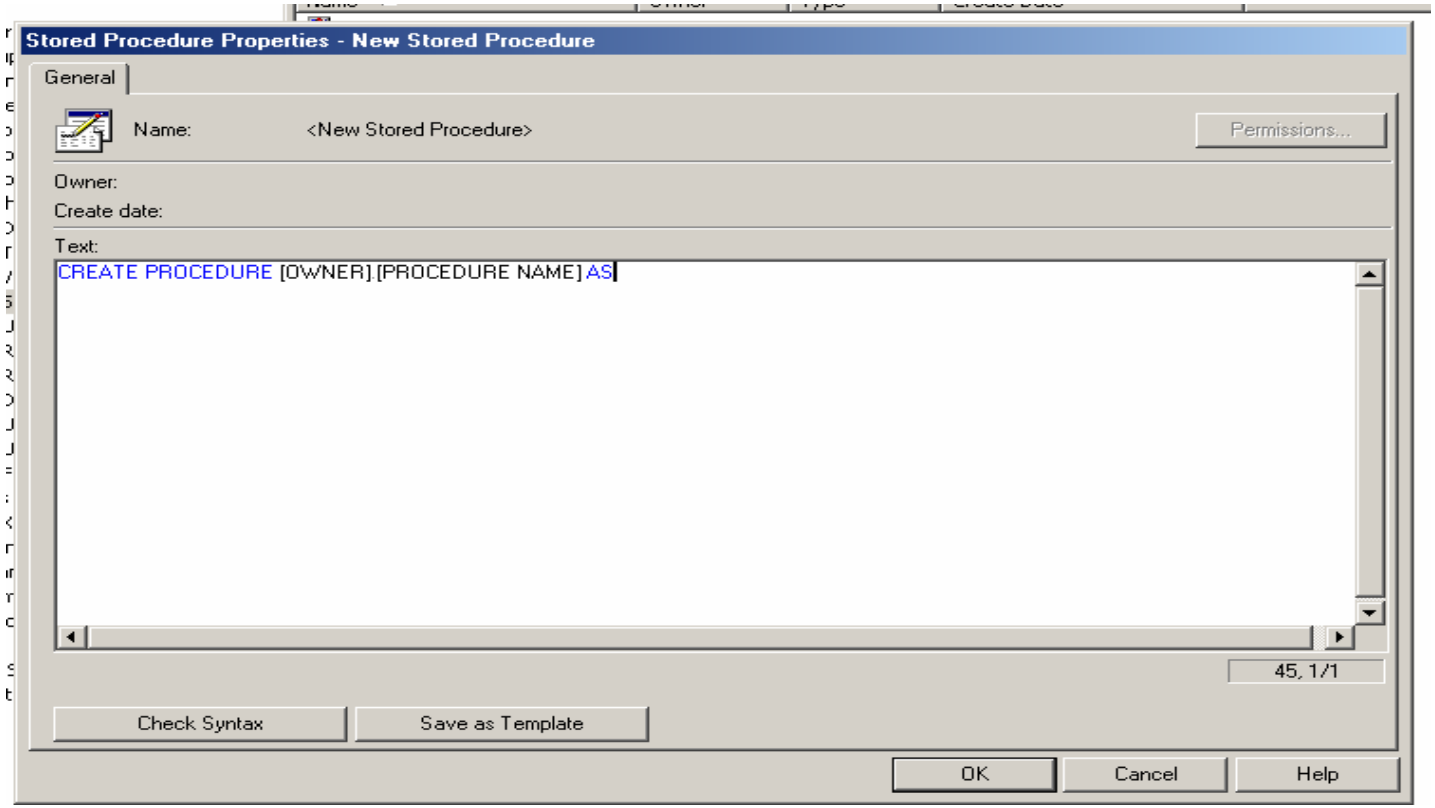
- Câu lệnh In trong CASE
- Hàm Cast chuyển đổi giá trị
- Hàm Left, right lấy chuỗi ký tự con
- Hàm str() chuyển giá trị về dạng ký tự
- Toán tử % lấy giá trị dư của phép chia

Xây dựng Thủ tục thường trú bằng Enterprise Manager

- Chúng ta có thể xây dựng thủ tục thường trú bằng 2 cách, sử dụng Câu lệnh Create và dùng giao diện Enterprise Manager.
- Dù tạo bằng cách nào, thủ tục thường trú cũng trở thành một đối tượng của CSDL.
- Chúng ta cũng có thể kiểm tra các đối tượng băng Enterprise Manager

Tạo Thủ Tục Thường Trú Bằng EM

Đưa chuột đến Mục Stored Procedure nhấp chuột phải chọn New Stored Procedure



Tạo Thủ Tục Thường Trú Bằng EM

Màn hình soạn thảo sẽ như hình trên.

Nút Check Syntax là công cụ giúp chúng ta kiểm tra cú pháp của các câu lệnh trong thủ tục thường trú.

Sau khi soạn thảo chúng ta chỉ cần check Syntax để kiểm tra cú pháp trước khi lưu

Thay đổi thủ tục thường trú

Cú pháp:

```
Alter Procedure Ten_thutuc
[<cac tham so>]
[<cac gia tri mac dinh>]
As
Begin
   caulenh_sql1
   caulenh_sql2
End
```

Thay đổi thủ tục thường trú trong Enterprise Manager

Khi thay đổi trong Enterprise Manager chúng ta chỉ cần nhấp đúp chuột vào thủ tục thường trú muốn thay đổi

Nội dung của thủ tục thường trú được hiển thị

Check Syntax trước khi lưu

Xoá thủ tục thường trú

Kiểm tra chắc chắn các thông tin của thủ tục trước khi xoá, kể cả về nội dung và ảnh hưởng của thủ tục đó đối với hệ thống.

Cú pháp:

Drop Procedure ten_thutuc

Xoá thủ tục thường trú trong Enterprise Manager

Khi xoá thủ tục thường trú trong Enterprise Manager ta làm như sau:

Đưa chuột đến thủ tục thường trú rồi nhấp chuột phải tại thủ tục cần xoá.

Chọn Delete trong menu

Thủ tục sẽ bị xoá nếu người dùng có quyền được xoá Thủ tục thường trú

Một Số
STORED PROCEDURE
Của Hệ Thống

Một số Stored Procedure của hệ thống

- Trong cơ sở dữ liệu SQL server có rất nhiều Stored Procedure hệ thống. Chúng ta có thể gọi những thủ tục thường trú này khi cần thiết.

- Trong một tình huống nào đó, giả sử cần đến các thông tin liên quan đến hệ thống SQL server hay CSDL, những thủ tục này sẽ giúp ích rất nhiều cho mục đích của chúng ta

Một số Stored Procedure của hệ thống

Sp_who2;

Trả về liệt kê ra tất cả các người dùng truy cập đến SQL server hay cơ sở dữ liệu

Sp_configure;

Muốn biết cấu hình của SQL server 2000 đang dùng.

Một số Stored Procedure của hệ thống

- Sp_tables: Cung cấp thông tin về bảng
- Sp_password: cho phép thay đổi password của một user
- Sp_helptext: hiển thị nội dung của View và thủ tục thường trú
- Sp_helpdb: hiển thị nội dung của cơ sở dữ liệu.
- Sp_dropuser: Xoá một user
- Sp_columns: hiển thị các cột của bảng
- Sp_adduser: thêm một user vào CSDL
- Sp_addlogin: tạo một user

Một số bảng chứa các thông tin hệ thống

Thông tin hệ thống

- Trong SQL server có một bảng chứa các thông tin hệ thống SQL server hay thông tin của các cơ sở dữ liệu thành phần.

- Bên cạnh đó còn có các bảng chứa các thông tin về những đối tượng của CSDL.

Thông tin hệ thống

Syslogins: chứa danh sách người dùng trong cơ sở dữ liệu SQL Server

Sysobjects: Chứa danh sách các đối tượng trong CSDL SQL Server

Chú ý:

Chúng ta cần xem xét kỹ khi sử dụng thông tin hệ thống. Đặc biệt khi cập nhật dữ liệu vào thông tin hệ thống

Xử lý lỗi trong thủ tục thường trú

Tìm hiểu các thủ tục thường trú của hệ thống

Xử lý lỗi trong thủ tục thường trú và các thủ tục hệ thống

- Khái niệm cơ bản về Lỗi (Error)
- Kiểm soát và thay đổi nội dung lỗi
- Lỗi trong SQL server
- Một số Thủ tục hệ thống

Khái niệm cơ bản về Lỗi

- Lỗi phát sinh khi bảng, Biến hay tham số không chấp nhận giá trị truyền vào. Một trong những yếu tố không hợp lệ xảy ra đều khiến SQL server phát sinh ra lỗi.
- Lỗi được đưa ra màn hình có nội dung đã được SQL server định nghĩa trước.
- Tuy nhiên, nếu cần chúng ta có thể thay đổi nội dung thông báo sao cho phù hợp với yêu cầu của người sử dụng

Xử lý lỗi

- Trong khi lập trình trên SQL server, sẽ có lúc chúng ta cần kiểm soát lỗi do các câu lệnh SQL gây ra.
- Một số lỗi do sai kiể dữ liệu, một số lỗi do phân quyền truy cập không hợp lệ…
- Khi phát sinh lỗi, nếu cần chúng ta có thể bắt lỗi và kiểm soát chúng.

Xử lý lỗi

- ShipperID là cột không cho phép nhập dữ liệu
- Do vậy khi thêm một bản ghi mới và gán dữ liệu cho cột ShipperID thì không thêm được ➔ Sảy ra lỗi
  - Declare @Error int
  - Insert into Shippers(shipperID) Values(1)
  - Select @Error = @@ERROR
  - print 'Gia tri bien loi la :' + convert(varchar,@Error)
  - print 'Gia tri ham loi la :' + convert(varchar,@@Error)

Sử dụng @ERROR trong thủ tục thường trú

```
Create procedure sp_XulyLoi
AS
Begin
 Declare @Error int
 Insert into Shippers(shipperID) Values(1)
 set @Error = @@ERROR
 if @Error !=0
  Begin
   if @Error=544
    Begin
      print 'Khong them ban ghi moi duoc'
      print 'Cot ShipperID khong thay doi duoc.'
    End
   Else
      print 'Khong phan biet duoc loi'
      print 'Lien lac voi Adminnistrator'
      print 'Loi co so:' + convert(varchar,@Error)
  End
End
```

Sử dụng @ERROR trong thủ tục thường trú

Thủ tục trên sẽ in ra thông tin khi có lỗi:

In ra thông tin trước ELSE khi lỗi là lúc thêm mới vào bản ghi trong trường hợp xác định là cập nhật dữ liệu của cột ShipperID

In ra thông tin sau ELSE khi lỗi nhưng chưa xác định lỗi là do nguyên nhân tư đâu.

Kiểm soát lỗi khi sảy ra

Đây là thủ tục lấy mã lỗi thi thêm mới một bản ghi vào cơ sở dữ liệu

```
Create procedure sp_XulyLoiTraVe
AS
Begin
  Declare @Error int
  Insert into Shippers(shipperID) Values(1)
  set @Error = @@ERROR
  Return @Error
End
```

Kiểm soát lỗi khi sảy ra

```
Create procedure sp_InNoiDungLoi
@vError int
AS
Begin
 if @vError !=0
  Begin
   if @vError=544
    Begin
      print 'Khong them ban ghi moi duoc'
      print 'Cot ShipperID khong thay doi duoc.'
    End
   Else
    Begin
      print 'Khong phan biet duoc loi'
      print 'Lien lac voi Adminnistrator'
      print 'Loi co so:' + convert(varchar,@vError)
    End
 End
End
```

Kiểm soát lỗi khi sảy ra

Thủ tục trên sẽ chuyển mã lỗi thành ngôn ngữ của người sử dụng.

Giá trị truyền vào là mã lỗi.

Những thông tin được in ra tương ứng với những mã lỗi được truyền vào.

Kết hợp hay thủ tục trên

Kết hợp hai thủ tục trên ta có kết quả như mong muốn: Thông báo lỗi theo ngôn ngữ của người lập trình (yêu cầu của thiết kế)

- Declare @Error int
- Exec @Error = sp_XulyLoiTraVe
- exec sp_InNoiDungLoi @Error

- Lấy giá trị của thủ tục thứ nhất ra ngoài.
- Truyền vào trong thủ tục 2

Lệnh RaisError

Trong khi SQL server sử dụng lệnh RaisError để đưa thông báo lỗi ra màn hình.

Tuy nhiên ở mức độ nào đó chúng ta cũng có thể dùng lệnh này để phát sinh lỗi theo nội dung mong muốn.

Cú pháp Lệnh RaisError

RAISERROR (

<ID thông báo | Nội dung thông báo>, <Chỉ dẫn phát sinh lỗi>, <trạng thái>

[,các tham số]

)

Trạng thái (state): nhằm chỉ rõ lỗi thuộc nhóm nào trong hệ thống của SQL server, trạng thái có giá trị từ: 1 đến 27

ID thông báo, Nội dung (message ID, Message String)

ID thông báo là số ID của nội dung lỗi trong hệ thống lỗi của SQL Server.

Các thông báo lỗi của hệ thống chứa trong bảng sysmessages.

Chúng ta có thể thêm thông báo lỗi vào bảng hệ thống bằng cách sử dụng Stored Procedure

Chỉ dẫn phát sinh lỗi (Severity)

- Là code chỉ dẫn phát sinh lỗi.
- Trong SQL Server nếu lỗi phát sinh do dữ liệu thì Severity có giá trị từ 1-18.
- Nếu lỗi xuất phát từ hệ thống thì severity có giá trị từ 20-25

Ví dụ RAISERROR

```
Create procedure sp_XulyLoiRais
AS
Begin
 Declare @Error int
 Insert into Shippers(shipperID) Values(1)
 set @Error = @@ERROR
 if @Error !=0
  Begin
   Raiserror('Loi them du lieu',1,1)
  End
End
```

Thêm thông báo lỗi vào danh sách lỗi

Hầu hết các lỗi sảy ra khi phát sinh trong quá trình làm việc trên CSDL SQL Server.

Trong một số trường hợp nào đó chúng ta cần có những thông báo lỗi như nội dung mình mong muốn

Cú pháp thêm thông báo lỗi

sp_addmessage [@msgnum =] *msg_id* ,
[@severity =] *severity* ,
[@msgtext =] '*msg*'
[, [@lang =] '*language*']
[, [@with_log =] '*with_log*']
[, [@replace =] '*replace*']

Một số tham số đã trình bày trên RAISERROR

Thủ tục mở rộng
Extended Stored Procedure

Thủ tục mở rộng

xp_cmdshell: Thực hiện lệnh dưới dấu nhắc DOS từ SQL server

Cú pháp:

Xp_cmdshell <‘Câu lệnh’> [, no_output]

Thủ tục mở rộng

Xp_msver: trả về thông tin nơi SQL server cài đặt.

Nói chung còn rất nhiều thủ tục mở rộng trong SQL server mà chúng ta chưa tìm hiểu.

Để tìm hiểu các thủ tục mở rộng chúng ta có thể vào Extended Stored Procedure trong Enterprise Manager.

Các thủ tục khác của hệ thống