Tài liệu hướng dẫn: Stored Procedure

Người thực hiện

– Hoàng Anh Tú

Nội dung

1 Giới thiệu 2

2 Stored Procedure cơ bản 2

2.1 Tạo Stored Procedure 3

2.1.1 Tạo bằng SQL Server Management Studio (SSMS) 3

2.1.2 Tạo bằng script 4

2.2 Thực thi Stored Procedure 4

2.2.1 Thực thi bằng giao diện SSMS 4

2.2.2 Thực thi bằng script 4

2.3 Thay đổi nội dung Stored Procedure 5

2.3.1 Thực thi bằng script 5

2.4 Xóa Stored Produre 5

2.4.1 Thực thi bằng script 5

3 Tham số trong Stored Procedure 5

3.1 Tham số đầu vào 5

3.2 Tham số đầu ra 6

4 Trả về giá trị trong Stored Procedure 7

4.1 Trả về giá trị từ lệnh RETURN 7

4.2 Trả về dữ liệu từ lệnh SELECT 8

5 Kết hợp Stored Procedure với các lệnh T-SQL 9

5.1 Ứng dụng thêm sinh viên vào cơ sở dữ liệu 9

5.2 Ứng dụng trả về danh sách sinh viên trong lớp 9

1 Giới thiệu

Tài liệu này hướng dẫn thực hành Stored Procedure trong Hệ quản trị cơ sở dữ liệu SQL Server. Nội dung chính bao gồm những phần sau:

- Hướng dẫn cú pháp Stored Procedure trong SQL Server.
- Cách gọi, sử dụng lại các Stored Procedure đã tạo.
- Một số ví dụ Stored Procedure.

2 Stored Procedure cơ bản

Khi chúng ta tạo một ứng dụng với Microsoft SQL Server, ngôn ngữ lập trình T-SQL (Transact-SQL) là ngôn ngữ chính giao tiếp giữa ứng dụng và database của SQL Server. Khi chúng ta tạo các chương trình bằng T-SQL, hai phương pháp chính có thể dùng để lưu trữ và thực thi cho các chương trình là:

- Chúng ta có thể lưu trữ các chương trình cục bộ và tạo các ứng dụng để gởi các lệnh đến SQL Server và xử lý các kết quả.
- Chúng ta có thể lưu trữ những chương trình như các stored procedure trong SQL Server và tạo ứng dụng để gọi thực thi các stored procedure và xử lý các kết quả.

Đặc tính của Stored-procedure trong SQL Server:

- Stored Procedure là hàm cho phép truyền tham số vào và trả về giá trị.
- Bao gồm 1 tập các lệnh T-SQL để xử lý 1 chức năng nào đó trong cơ sở dữ liệu.

Ta có thể dùng T–SQL EXECUTE để thực thi các stored procedure. Stored procedure khác với các hàm xử lý(User-defined Function) là giá trị trả về của chúng không chứa trong tên và chúng không được sử dụng trực tiếp trong biểu thức.

Stored procedure có những thuận lợi so với các chương trình T-SQL lưu trữ cục bộ là:

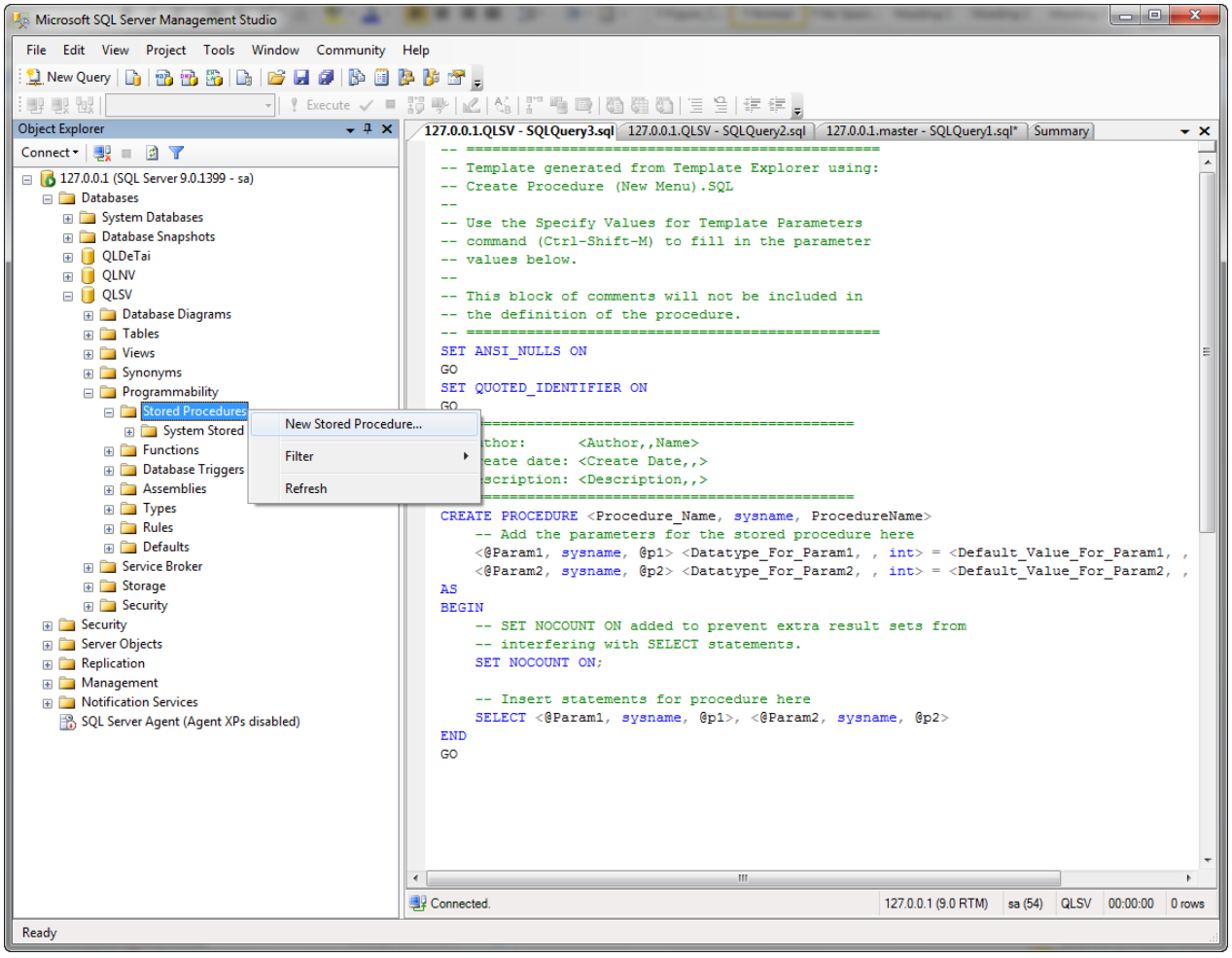
- ***Stored procedure cho phép điều chỉnh chương trình cho phù hợp:*** Chúng ta có chỉ tạo stored procedure một lần và lưu trữ trong database một lần, trong chương trình chúng ta có thể gọi nó với số lần bất kỳ. Stored procedure có thể được chỉ rõ do một người nào đó tạo ra và sự thay đổi của chúng hoàn toàn độc lập với source code của chương trình.
- ***Stored procedure cho phép thực thi nhanh hơn:*** nếu sự xử lý yêu cầu một đoạn source code Transact – SQL khá lớn hoặc việc thực thi mang tính lặp đi lặp lại thì stored procedure thực hiện nhanh hơn việc thực hiện hàng loạt các lệnh Transact-SQL. Chúng được phân tích cú pháp và tối ưu hóa trong lần thực thi đầu tiên và một phiên bản dịch của chúng trong đó sẽ được lưu trong bộ nhớ

để sử dụng cho lần sau, nghĩa là trong những lần thực hiện sau chúng không cần phải phân tích cú pháp và tối ưu lại, mà chúng sẽ sử dụng kết quả đã được biên dịch trong lần đầu tiên.

- ***Stored procedure có thể làm giảm bớt vấn đề kẹt đường truyền mạng:*** giả sử một xử lý mà có sử dụng hàng trăm lệnh của Transact-SQL và việc thực hiện thông qua từng dòng lệnh đơn, như vậy việc thực thông qua stored procedure sẽ tốt hơn, vì nếu không khi thực hiện chúng ta phải gởi hàng trăm lệnh đó lên mạng và điều này sẽ dẫn đến tình trạng kẹt mạng.
- ***Stored procedure có thể sử dụng trong vấn đề bảo mật của máy:*** vì người sử dụng có thể được phân cấp những quyền để sử dụng các stored procedure này, thậm chí họ không được phép thực thi trực tiếp những stored procedure này.

2.1 Tạo Stored Procedure

2.1.1 Tạo bằng SQL Server Management Studio (SSMS)



Hình 1. Tạo Stored Procedure bằng chương trình SSMS

Tạo bằng giao diện SSMS sẽ phát sinh script tạo Stored Procedure sẵng, ta chỉ cần thêm nội dung vào trong phần thân hàm.

2.1.2 Tạo bằng script

Cú pháp đơn giản:

```
CREATE PROCEDURE  procedure_name
     @parameter1 data_type      [output] /*các tham số*/,
     @parameter2 data_type      [output]
AS
BEGIN
        [khai báo các biến cho xử lý]
        {Các câu lệnh transact-sql}
END
GO
```

Phần [output] là phần có thể có hoặc không để xác định loại tham số.

Ví dụ:

```
CREATE PROCEDURE  XinChao
     @hoTen    nvarchar(50)
AS
BEGIN
     print N'Xin chào ' + @hoTen
END
GO

CREATE PROC Hello
AS
BEGIN
     print N'Hello ' + @hoTen
END
GO
```

2.2 Thực thi Stored Procedure

2.2.1 Thực thi bằng giao diện SSMS

2.2.2 Thực thi bằng script

Sử dụng lệnh EXECUTE (có thể viết tắt là EXEC) để thực thi một stored procedure.

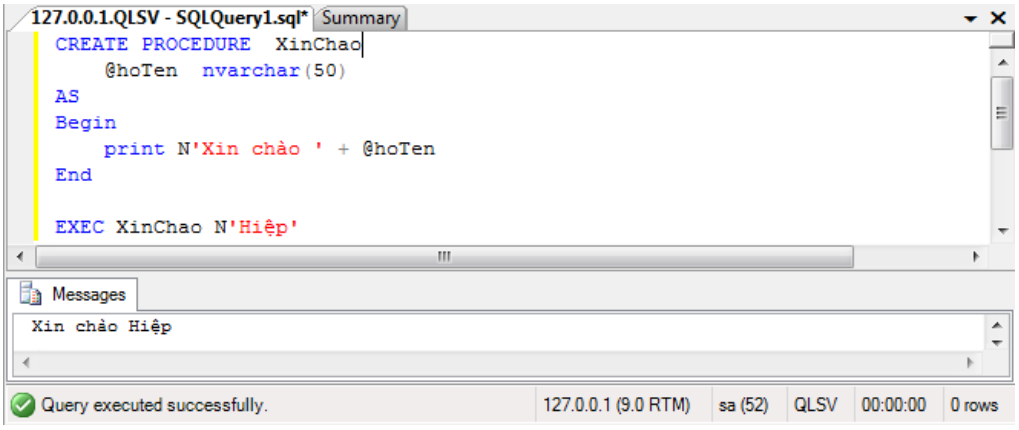
```
EXECUTE procedure_name   parameter_value1, parameter_value2,..

EXEC procedure_name parameter_value1, parameter_value2, ...
```

Ví dụ:

```
EXEC XinChao N'Hiệp'
```

Đoạn lệnh trên sẽ tạo kết quả như sau:



Hình 2. Kết quả thực thi Stored Procedure "XinChao"

2.3 Thay đổi nội dung Stored Procedure

2.3.1 Thực thi bằng script

```
ALTER PROCEDURE  procedure_name
     @parameter1 data_type      [output] /*các tham số*/,
     @parameter2 data_type      [output]
AS
BEGIN
        [khai báo các biến cho xử lý]
        {Các câu lệnh transact-sql}
END
GO
```

Lúc này, SQL Server sẽ thay thế stored procedure có tên "procedure_name" bằng 1 stored procedure mới có cùng tên.

2.4 Xóa Stored Produre

2.4.1 Thực thi bằng script

```
DROP PROCEDURE  procedure_name

DROP PROC procedure_name
```

3 Tham số trong Stored Procedure

Stored Procedure là 1 hàm được lưu trữ sẵng trong cơ sở dữ liệu. Hàm này có thể có 2 loại tham số chính: tham số đầu vào và tham số đầu ra.

3.1 Tham số đầu vào

Đây là loại tham số mặc định, cho phép truyền các giá trị vào trong stored procedure để hỗ trợ xử lý.

Ví dụ:

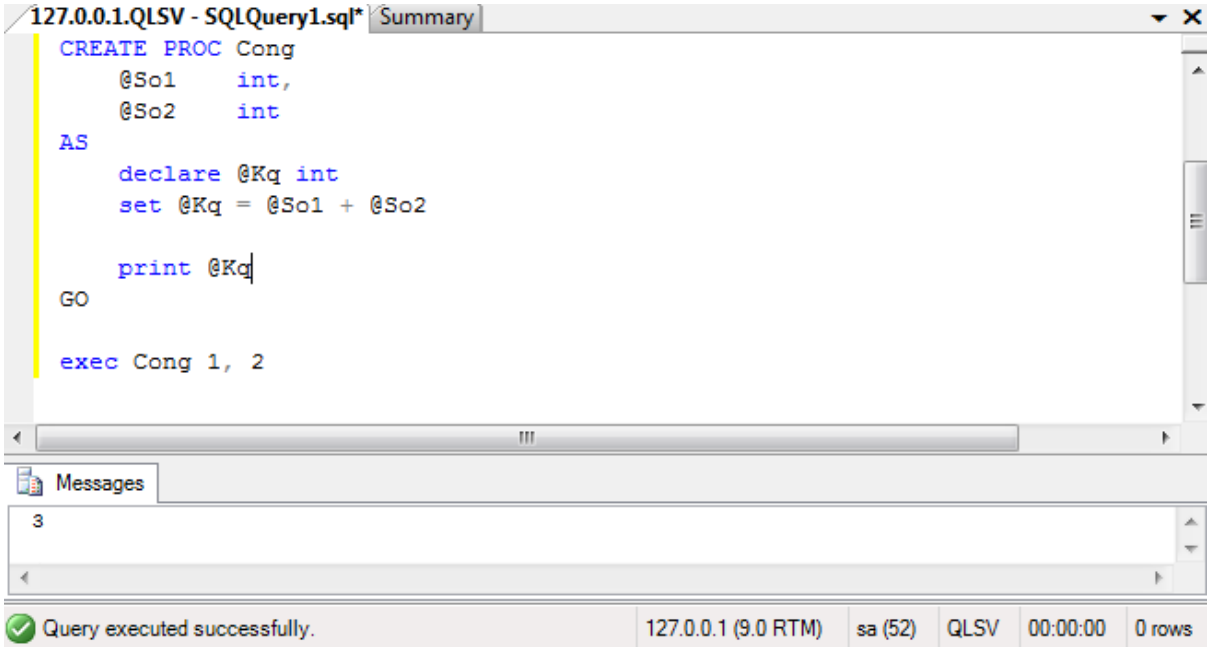
```
CREATE PROC Cong
```

```
      @So1 int,
      @So2 int
AS
BEGIN
      declare @Kq int
      set @Kq = @So1 + @So2

      print @Kq
END
GO

exec Cong 1, 2
```

Kết quả đoạn lệnh trên sẽ cho kết quả là "3"



Hình 3. Kết quả thực thi stored procedure cộng 2 số nguyên

3.2 Tham số đầu ra

Tham số dùng để nhận kết quả trả về từ stored procedure. Sử dụng từ khóa OUTPUT (hoặc viết tắt là OUT) để xác định tham số.

Ví dụ:

```
ALTER PROC Tru
      @So1 int,
      @So2 int,
      @Kq  int output
AS
BEGIN
      set @Kq = @So1 - @So2
END
GO
```

```
DECLARE @test int
EXEC Tru 1, 2, @test output
PRINT @test
```

Kết quả in ra màn hình là “-1”.

4 Trả về giá trị trong Stored Procedure

Ngoài cách sử dụng tham số đầu ra để trả về giá trị. Có thể sử dụng RETURN để trả về giá trị từ stored procedure hoặc các câu lệnh SELECT khi truy vấn dữ liệu.

4.1 Trả về giá trị từ lệnh RETURN

Lệnh RETURN được sử dụng để trả về giá trị từ stored procedure mà không cần sử dụng tham số đầu ra. Giá trị trả về này có một số đặc điểm:

- Giá trị trả về chỉ có thể là số nguyên. Nếu trả về các loại giá trị khác thì lúc thực thi stored procedure sẽ báo lỗi (ngoại trừ 1 số kiểu dữ liệu được tự động chuyển đổi sang kiểu số nguyên như:float, double,...).
- Giá trị trả về mặc định là 0.
- Có thể nhận giá trị trả về này bằng 1 biến.
- Sau khi gọi RETURN, stored procedure sẽ trả về giá trị và kết thúc xử lý.

Ví dụ:

```
CREATE PROC Test
     @Lenh     int
AS
BEGIN
     if (@Lenh = 1)
          return 1

     if (@Lenh = 2) begin
          declare @float float
          set @float = 2.6

          return @float
     end

     if (@Lenh = 3) begin
          declare @char varchar(50)
          set @char = 'hello'

          return @char
     end
END
GO
```

```
declare @test float
EXEC @test = Test 3
print @test
```

Nếu giá trị truyền vào là 1: stored procedure trả về giá trị “1”.

Nếu giá trị truyền vào là 2: stored procedure trả về giá trị “2”.

Nếu giá trị truyền vào là 3: stored procedure báo lỗi không thể chuyển chuỗi ‘hello’ thành số nguyên.

Nếu truyền các giá trị khác: stored procedure trả về giá trị “0”.

4.2 Trả về dữ liệu từ lệnh SELECT

Mỗi lệnh SELECT đặt trong stored procedure sẽ trả về 1 bảng.

```
CREATE PROC TestSelect
AS
BEGIN
     SELECT * FROM SINHVIEN

     SELECT * FROM LOP
END
GO

EXEC TestSelect
```

Kết quả in ra màn hình sẽ là:

Results | Messages

	ma	hoTen	namSinh	danT...	maLop
1	0212001	Nguyễn Vĩnh An	1984	Kinh	TH2002/01
2	0212002	Nguyễn Thanh Bình	1985	Kinh	TH2002/01
3	0212003	Nguyễn Thanh Cường	1984	Kinh	TH2002/02
4	0212004	Nguyễn Quốc Duy	1983	Kinh	TH2002/02
5	0312001	Phan Tuấn Anh	1985	Kinh	VL2003/01
6	0312002	Huỳnh Thanh Sang	1984	Kinh	VL2003/01

	ma	maKhoaHoc	maKhoa	maChuongTrinh	soThuTu
1	TH2002/01	K2002	CNTT	CQ	1
2	TH2002/02	K2002	CNTT	CQ	2
3	VL2003/01	K2003	VL	CQ	1

Query executed successfully. | 127.0.0.1 (9.0 RTM) | sa (51) | QLSV | 00:00:00 | 9 rows

Hình 4. Kết quả thực hiện stored procedure “TestSelect”

5 Kết hợp Stored Procedure với các lệnh T-SQL

Các stored procedure thông thường được tạo ra nhằm giúp thực hiện một số chức năng cần thao tác trong cơ sở dữ liệu. Khi đó, ta cần phải kết hợp nhiều lệnh T-SQL thao tác với dữ liệu như (SELECT, INSERT, UPDATE, DELETE) và các cấu trúc điều khiển (IF, WHILE, CASE,...).

5.1 Ứng dụng thêm sinh viên vào cơ sở dữ liệu

```
CREATE PROC ThemSinhVien
      @mssv       varchar(10),
      @hoTen      nvarchar(100),
      @namSinh    int,
      @danToc     nvarchar(20),
      @maLop      varchar(10)
AS
BEGIN
      IF(EXISTS(SELECT * FROM SinhVien s WHERE s.ma = @mssv)) BEGIN
            PRINT N'Mã số sinh viên ' + @mssv + N' đã tồn tại'
            RETURN -1
      END

      IF(NOT EXISTS(SELECT * FROM Lop L WHERE L.ma = @maLop)) BEGIN
            PRINT N'Mã số lớp ' + @maLop + N' chưa tồn tại'
            RETURN -1
      END

      INSERT INTO SinhVien(ma, hoTen, namSinh, danToc, maLop)
      VALUES(@mssv, @hoTen, @namSinh, @danToc, @maLop)
      RETURN 0 /* procedure tự trả về 0 nếu không RETURN */
END
GO

DECLARE @kq INT
EXEC @kq = ThemSinhVien '0212005', N'Nguyễn Văn A', 1987, 'Kinh',
'TH2002/01'
PRINT @kq
```

5.2 Ứng dụng trả về danh sách sinh viên trong lớp

```
CREATE PROC XuatDanhSachSinhVien
      @maLop     varchar(10)
AS
BEGIN
      IF(NOT EXISTS(SELECT * FROM Lop L WHERE L.ma = @maLop))
      BEGIN
            PRINT N'Mã số lớp ' + @maLop + N' chưa tồn tại'
            RETURN -1
      END

      SELECT * FROM Lop l where l.ma = @maLop
      /*procedure luôn trả về 0 nếu không RETURN*/
END
GO
```