**THỦ TỤC**

Trả lời.

1. **Biến cục bộ là:**

Biến chỉ sử dụng trong đoạn chương trình khai báo nó như Query Batch, Stored Procedure, Function , chứa giá trị thuộc một kiểu nhất định.

Biến cục bộ được bắt đầu bằng 1 kí hiệu: @

Ví dụ: @v\_a ; @v\_b; @v\_Result

1. **Khai báo**

DECLARE <@tên\_biến> <Kiểu\_dữ\_liệu>, ….

Ví dụ: Khai báo một biến

DELARE @v\_Result Int;

Khai báo một biến có giá trị 50

DECLARE @v\_a Int = 50;

Khai báo một biến có giá trị 100

DELARE @v\_b Int = 100;

1. **Gán giá trị cho biến.**

SET @tên\_biến = {giá\_trị | biến | biểu\_thức | SELECT ….}

Ví dụ: Tính tổng

SET @v\_Result = @v\_a + @v\_b;

1. **In ra màn hình – Lệnh PRINT**

Print @Biến

+ Khi có kết hợp với chuỗi , phải đổi kiểu dữ liệu sang kiểu chuỗi bằng hàm CAST hay CONVERT

Ví dụ: In ra màn hình Console( Dùng cho lập trình viên)

Sử dụng CAST để ép kiểu Int về kiểu chuỗi.

Sử dụng toán tử + để nối 2 chuỗi.

PRINT `v\_a =` + CAST(@v\_a AS varchar(15));

PRINT `v\_b =` + CAST(@v\_b AS varchar(15));

PRINT `v\_Result =` + CAST(@v\_Result AS varchar(15));

1. **Thủ tục (Store Procedure).**

Thủ tục là một đối tượng trong hệ quản trị CSDL bao gồm các câu lệnh SQL. Chúng được kết hợp lại với nhau thành một khối lệnh , dùng để thực hiện một số công việc nào đó như cập nhật, thêm mới, xóa, hiển thị ,tính toán và có thể trả về các giá trị.

**Lợi ích khi dùng Thủ tục.**

Chỉ cần viết thủ tục 1 lần, sau đó có thể gọi nó nhiều lần trong ứng dụng.

Thực thi nhanh hơn và giảm tải băng thông.

Việc cung cấp các Thủ tục cho người dùng không ảnh hưởng đến vấn đề bảo mật của SQL Server

1. **Có mấy loại thủ tục?**

Có 2 loại thủ tục:

**Thủ tục hệ thống** : là những thủ tục do SQL cung cấp ( tự nghiên cứu System Stored Procedure) . Tên có tiếp đầu ngữ *sp\_*

**Thủ tục người dùng** : do người dùng tạo ra, để dễ dàng phân biệt chúng ta quy định . Tên thủ tục có tiếp đầu ngữ *usp\_*

1. **Cú pháp khai báo thủ tục như thế nào?**

**CREATE PROCEDURE** <*Tên thủ tục* >

*Danh sách tham số vào*

*[Danh sách tham số ra <Output>]*

**AS**

**BEGIN**

**[ Khai\_báo\_các\_biến\_cục\_bộ\_của\_thủ\_tục]**

**< Đoạn chương trình xử lý>**

**END**

**GO**

Ví dụ : Một thủ tục có tham số.

CREATE PROCEDURE sp\_Product

@product\_id INT

AS

BEGIN

IF ( exists ( SELECT \* FROM Product WHERE product\_id = @product\_id ))

SELECT \* FROM Product WHERE product\_id = @product\_id

ELSE

PRINT N’ Không tìm thấy sản phẩm có mã ’ + str(@product\_id , 3);

END;

1. **Cú pháp để thực thi thủ tục là gì?**

**- Lời gọi thủ tục**

EXECUTE tên\_thủ\_tục [danh\_sách\_các\_đối\_số]

Số lượng các đối số cũng như thứ tự của chúng phải phù hợp với số lượng và thứ tự của các tham số khi định nghĩa thủ tục

Ví dụ: EXECUTE sp\_Product 6;

**- Chỉnh sửa thủ tục**

Thay từ khóa CREATE trong lệnh tạo thủ tục bằng từ khóa ALTER

*Ví dụ*: Giả sử tạo 1 thủ tục để xem thông tin của tất cả các Khách hàng (tham khảo bảng Customer đã tạo trước đó):

CREATE PROCEDURE sp\_ViewCustomer

AS

BEGIN

SELECT \* FROM Customer;

END;

* Bây giờ cần sử thủ tục trên để nó chỉ cho phép xem thông tin của Khách Hàng có năm sinh được nhập từ bàn phím, ta làm như sau:

ALTER PROCEDURE sp\_ViewCustomer (@year VARCHAR(4))

AS

BEGIN

SELECT \* FROM Customer WHERE datepart (year, Birth) = @year;

END;

*Thực thi :*

EXECUTE sp\_ViewCustomer 1980;

* Thủ tục sẽ được mã hóa nếu tùy chọn WITH ENCRYPTION được chỉ định.

Nếu thủ tục đã được mã hóa , ta không thể xem được nội dung của thủ

tục.

* Thêm từ khóa WITH ENCRYPTION trong lệnh ALTER thủ tục

*Ví dụ:* Nếu bây giờ ta muốn ngăn không cho người dùng xem thông tin của thủ tục sp\_ViewCustomer ta làm như sau:

ALTER TABLE PROCEDURE sp\_ViewCustomer( @year VARCHAR(4))

WITH ENCRYPTION

AS

BEGIN

SELECT \* FROM Customer WHERE datepart ( year, Birth) = @year;

END;

*Thực thi:*

EXCECUTE sp\_helptext `sp\_ViewCustomer`;

**- Xóa thủ tục**

**DROP PROCEDURE** tên\_thủ\_tục;

*Ví dụ:* DROP PROCEDURE sp\_ViewCustomer;

***Lưu ý*** : Bạn không xóa được thủ tục lưu trữ hệ thống nếu bạn không có quyền.

**- Biên dịch lại thủ tục**

Khi người sử dụng làm thay đổi tới những index của bảng. Thủ tục phải được biên dịch lại (recompiled) để chấp nhận những thay đổi đó.

Có **3** **cách** để biên dịch lại procedure:

* Sử dụng sp\_recompile system stored peocedure . Bạn có thể sử dụng cách này để biên dịch lại thủ tục ở lần chạy kế tiếp của nó.

Cú pháp:

sp\_compile [@objectname = ] ‘object’

* Chỉ ra WITH RECOMPILE trong câu lệnh CREATE PROCEDURE : SQL sẽ biên dịch lại thủ tục ở mỗi lần nó thực hiện.

Cú pháp:

CREATE PROCEDURE procedure\_name

@Parameter\_name data\_type

WITH RECOMPILE

AS

.

* Chỉ ra WITH RECOMPILE trong câu lệnh EXECUTE:

Biên dịch lại ngay ở lần thực hiện này.

Cú pháp:

EXECUTE procedure\_name WITH COMPILE

Source tham khảo: https://v1study.com/sql-thu-tuc-luu-tru-sp-a595.html