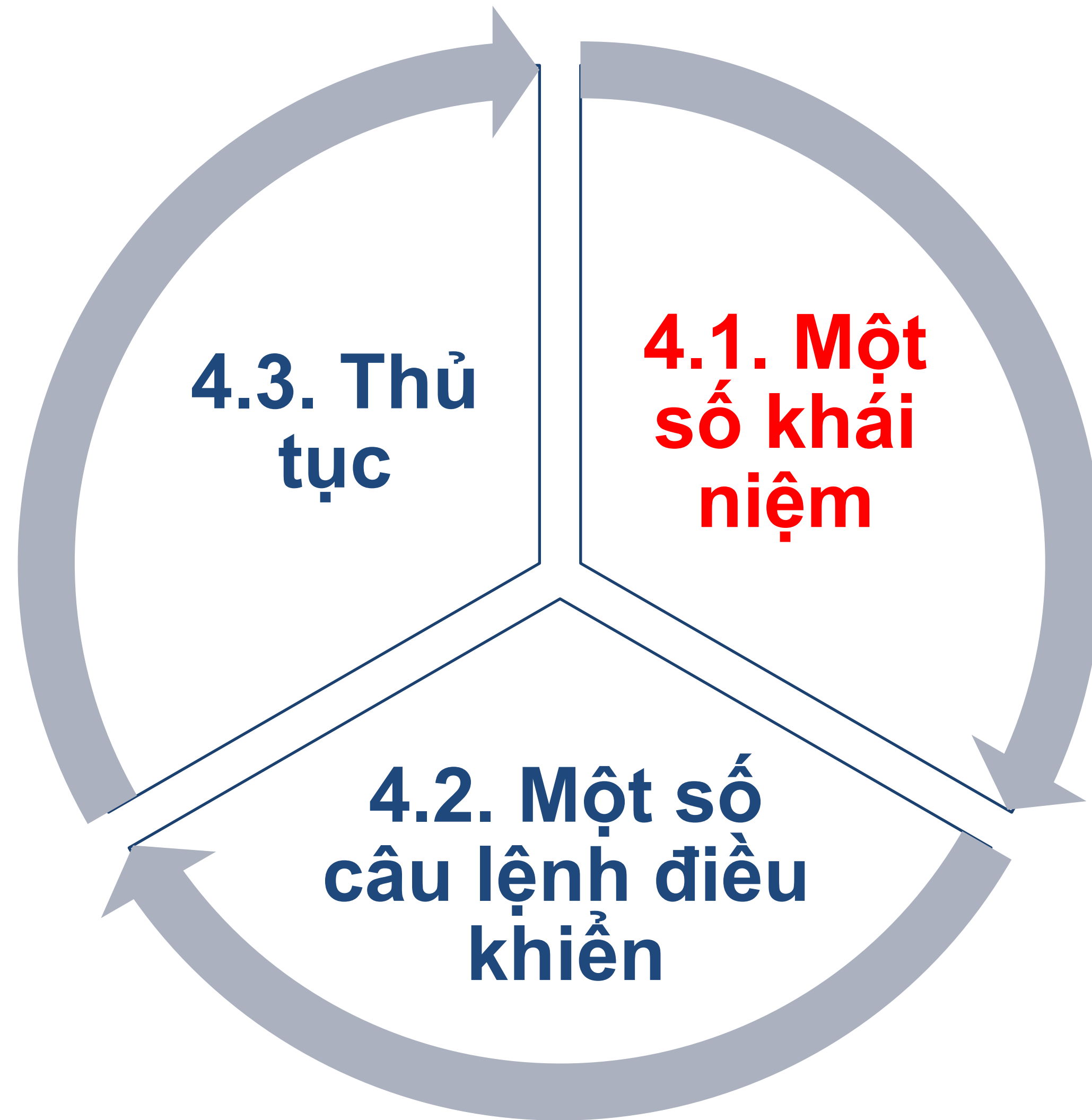




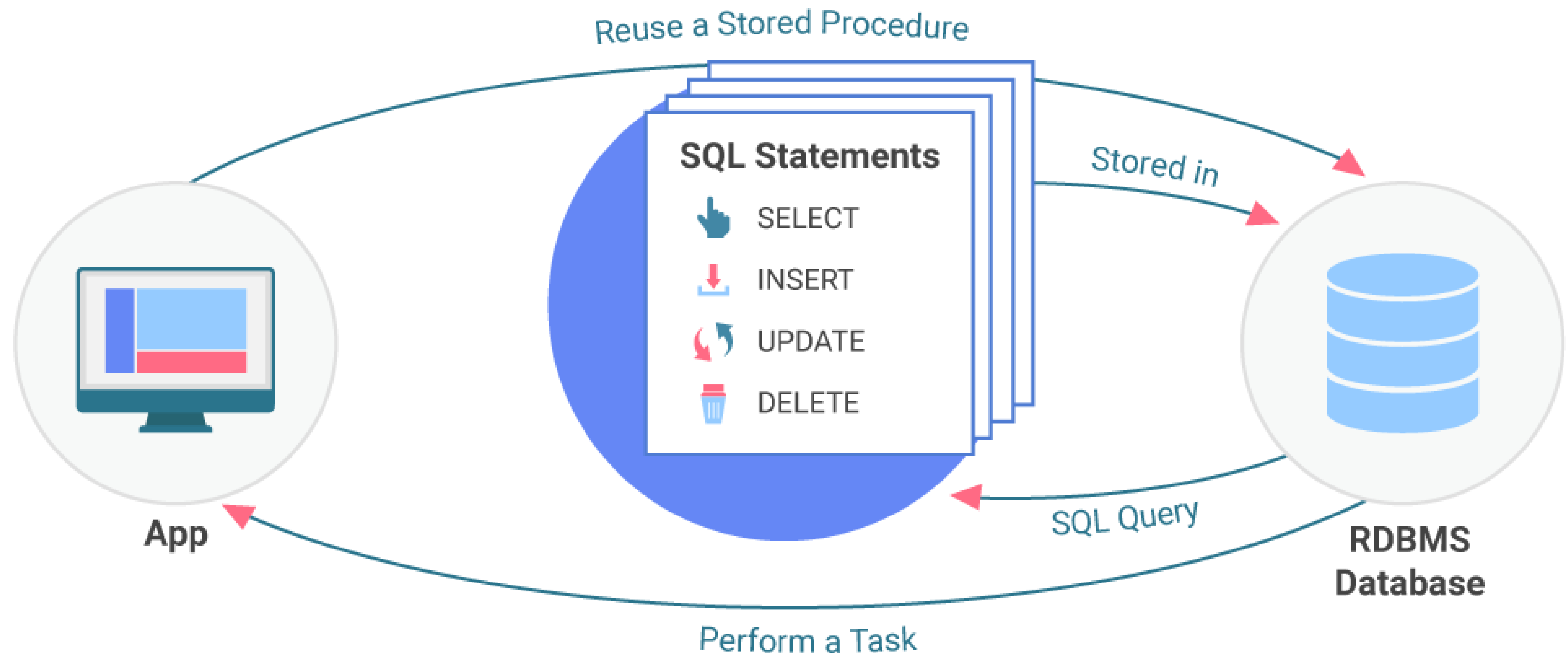
CHƯƠNG 4

THỦ TỤC

Giảng viên
ThS. Trần Thị Thanh Nhân



Stored Procedures



- Thủ tục lưu trữ là một đối tượng trong CSDL **gồm tập nhiều lệnh SQL được nhóm lại thành một nhóm** và **các lệnh này sẽ được thực hiện khi thủ tục lưu trữ được thực thi.**
- **Thủ tục lưu trữ có thể có các thành phần:**
 - Cấu trúc điều khiển (*IF, WHILE, FOR*)
 - Biến để lưu các giá trị tính toán, các giá trị truy xuất từ CSDL
 - Các câu lệnh SQL được kết hợp thành khối lệnh trong thủ tục, một thủ tục có thể có tham số truyền vào hay giá trị trả về giống như ngôn ngữ lập trình thông thường

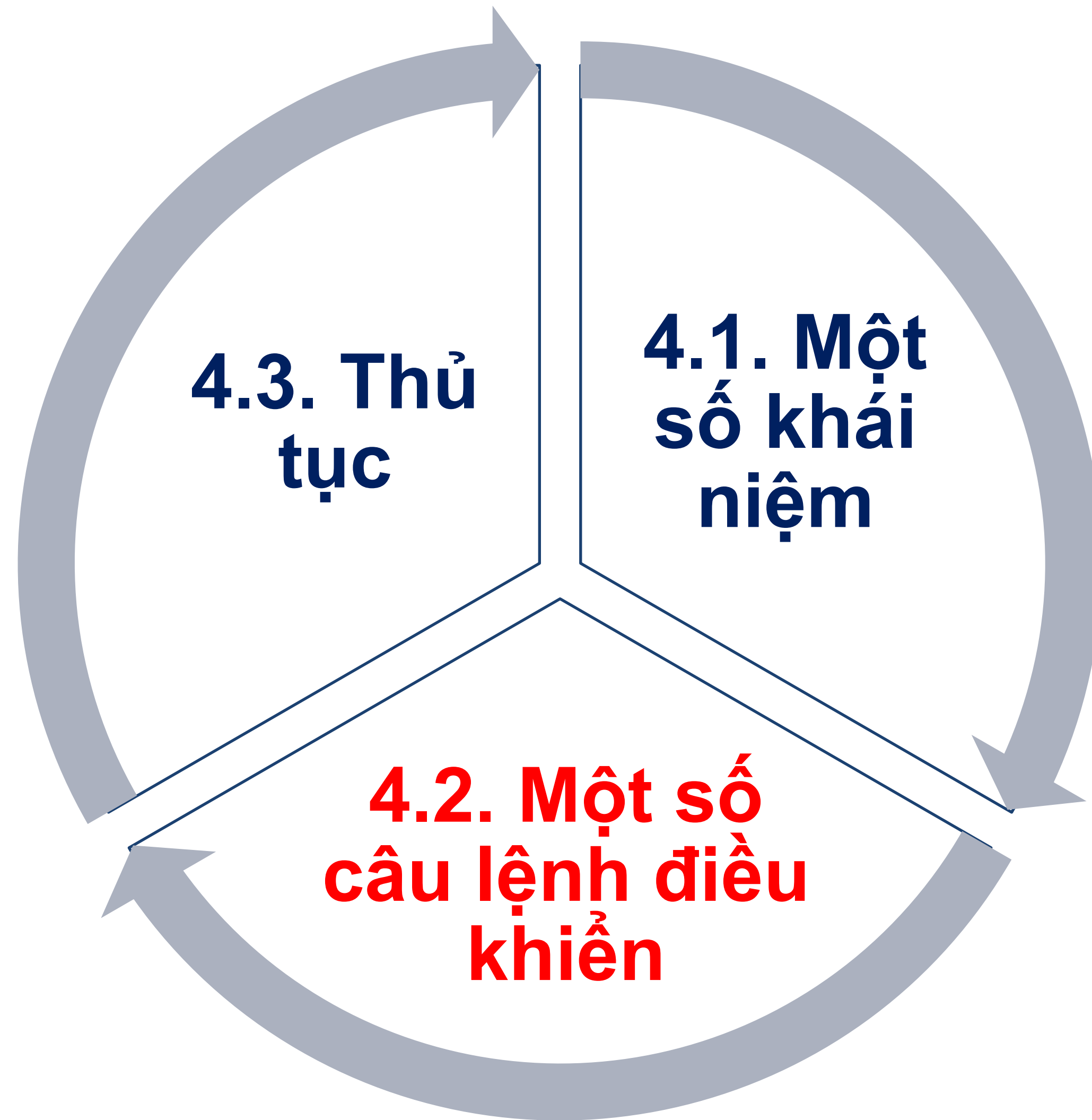
ƯU ĐIỂM CỦA STORED PROCEDURE

- Đơn giản hóa thao tác dữ liệu do tính module hóa thao tác
- Việc thực thi nhanh hơn so với thực hiện rời rạc các lệnh SQL thông thường
- Giảm lưu thông trên mạng: Thực hiện một yêu cầu bằng một câu lệnh đơn giản thay vì sử dụng nhiều dòng SQL
- Tăng bảo mật CSDL do việc cấp phát quyền trên thủ tục lưu thay vì tác động trực tiếp đến cơ sở dữ liệu
- Tập trung tại Server nên dễ quản lý
- Được biên dịch một lần và sử dụng lại kết quả trong các lần tiếp theo

- **System Stored Procedure:**
 - Lưu trữ trong CSDL Master
 - Bắt đầu bằng chữ **sp_***
 - Dùng trong quản trị CSDL và an ninh bảo mật.

VD: `sp_who @loginame = 'sa'`
`sp_server_info 1`
- **Local Stored Procedure:**
 - *Lưu trong CSDL do người dùng tạo ra*
 - *Được tạo bởi DBA (Database Administrator) hoặc người lập trình*
- **Remote Stored Procedure**
 - sử dụng thủ tục của một server khác

- **Temporary Stored Procedure**
 - Có chức năng tương tự Local Stored Procedure
 - Tự hủy khi kết nối tạo ra bị ngắt hoặc SQL Server ngừng hoạt động
 - Lưu trên CSDL TempDB
 - **Extended Stored Procedure:**
 - Sử dụng chương trình ngoại vi đã được biên dịch thành DLL.// các thư viện động Dynamic link library
 - Tên bắt đầu bằng **xp_***
- VD: xp_sendmail dùng gửi mail
- xp_cmdshell dùng thực hiện lệnh của DOS

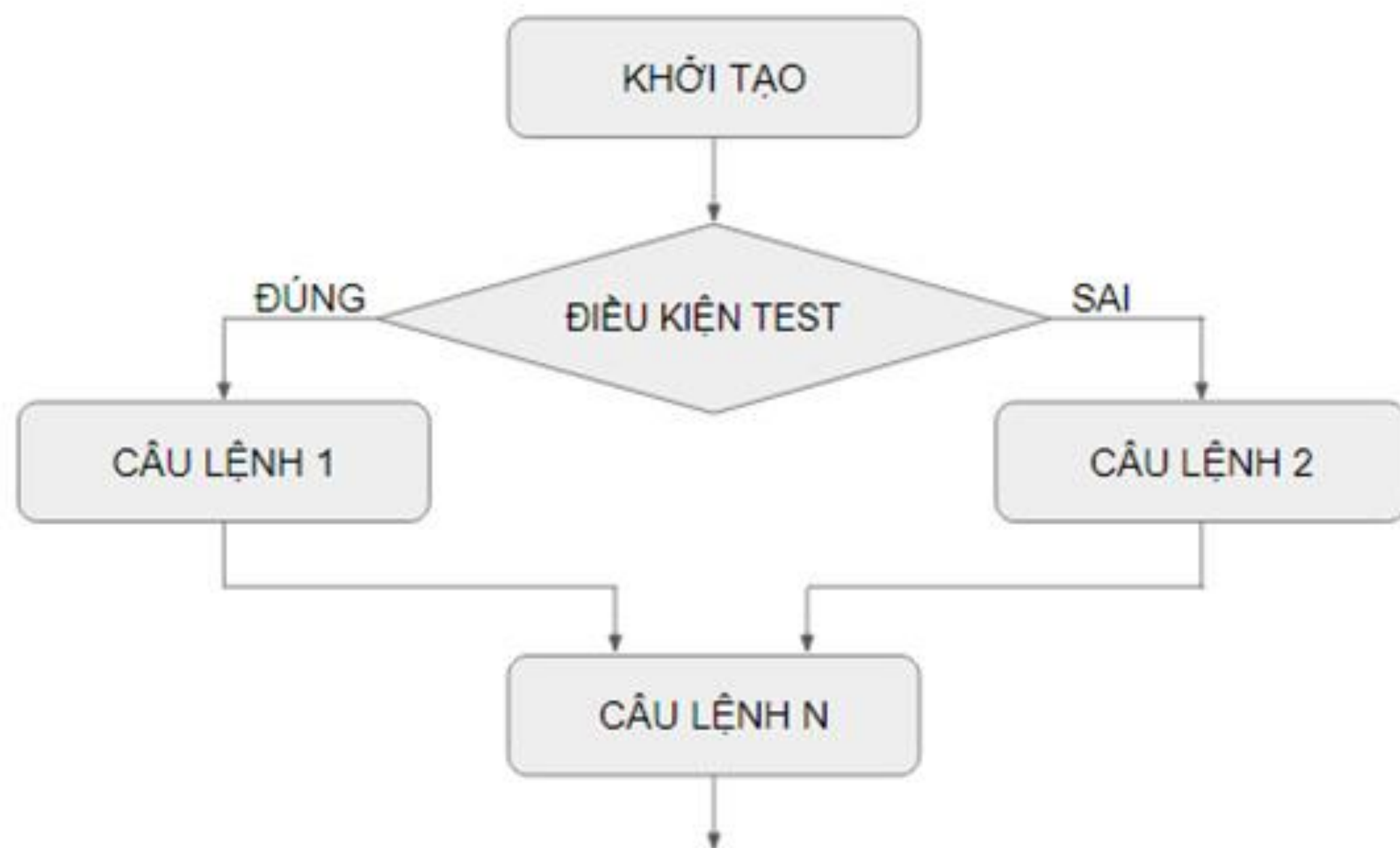


Cấu trúc rẽ nhánh If...else...

❑ Câu lệnh IF...ELSE dùng để **thực thi các lệnh có điều kiện**, nếu lệnh đúng thì thực thi lệnh đó, nếu sai sẽ thực thi một lệnh khác.

❑ **Cú pháp:**

IF dieukien {...câu lệnh thực thi khi điều kiện là TRUE...}
[ELSE {...câu lệnh thực thi khi điều kiện là FALSE...}]



❑ Vòng lặp WHILE là một lệnh luồng điều khiển được dùng để thực hiện **lặp lại các câu lệnh cho tới khi thỏa mãn điều kiện** cụ thể.

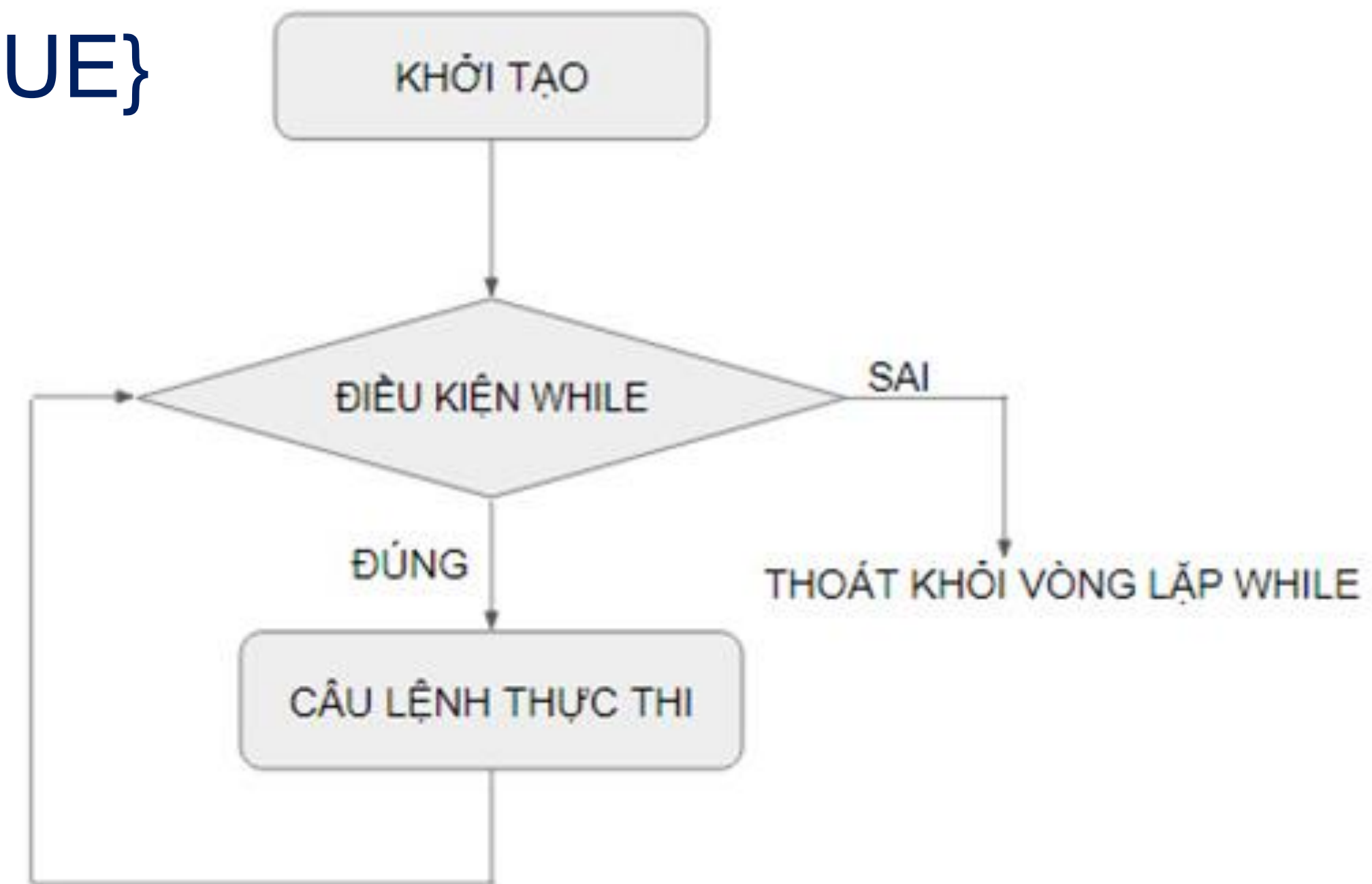
❑ **Cú pháp:**

WHILE dieukien /*các lệnh cần lặp*/

BEGIN

{câu lệnh thực thi khi điều kiện là TRUE}

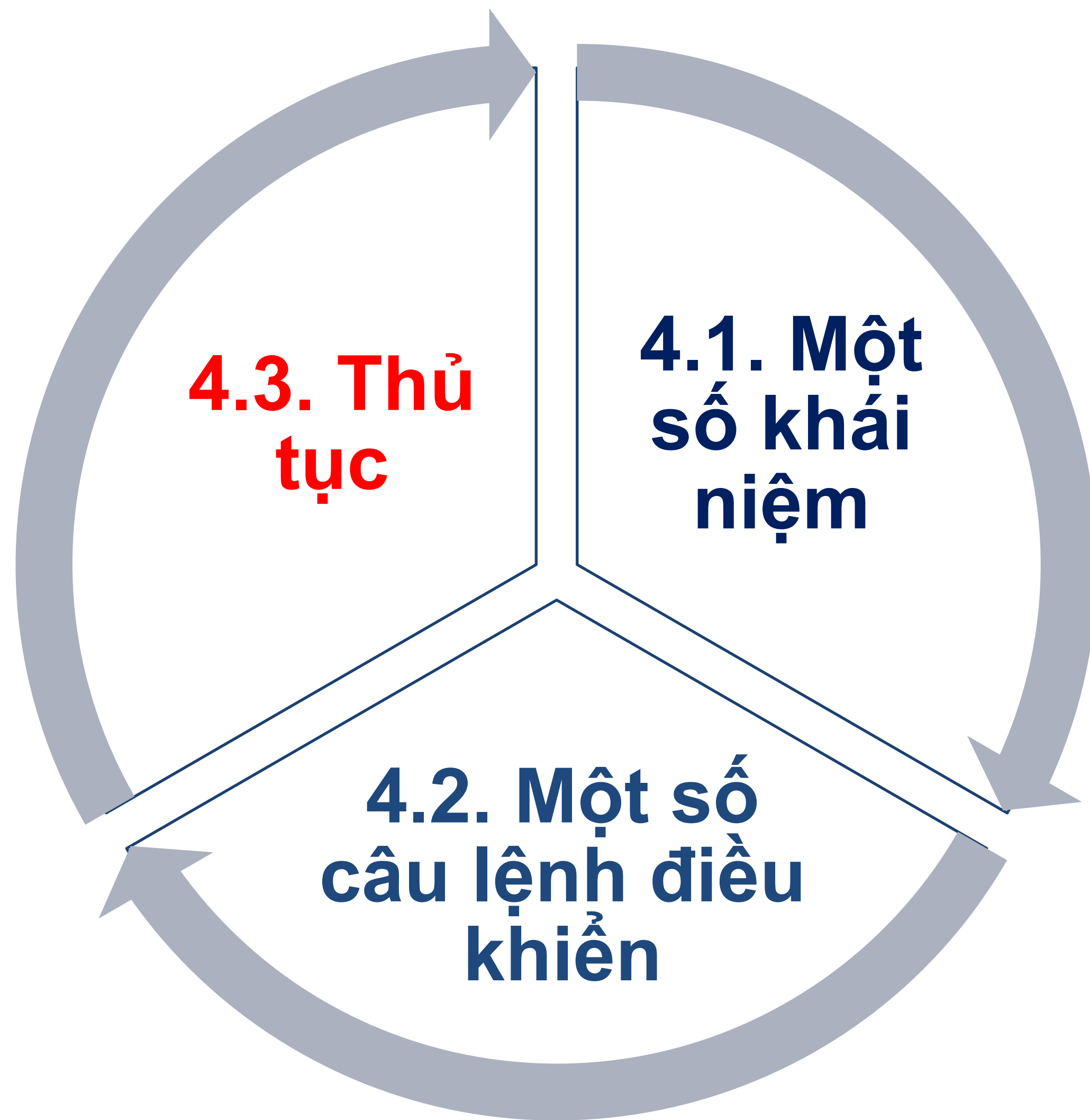
END;



- ❑ **Lệnh Break:** để thoát khỏi vòng lặp không xác định điều kiện **dừng** hoặc **muốn dừng** vòng lặp theo điều kiện do tự chỉ định và thực thi các câu lệnh tiếp sau lệnh END của vòng lặp.
- ❑ **Cú pháp:**

BREAK;
- ❑ **Ghi chú:**
 - ✓ Sử dụng BREAK để kết thúc sớm vòng lặp WHILE.
 - ✓ Nếu trong đoạn code có WHILE LOOP lồng nhau, BREAK sẽ chấm dứt vòng lặp WHILE gần nhất.

- ❑ **Lệnh CONTINUE** dùng để trao quyền thi hành lệnh cho biểu thức điều kiện của vòng lặp gần nhất.
- ❑ **Cú pháp**
CONTINUE;
- ❑ Ghi chú:
 - ✓ Không có tham số và đối số trong câu lệnh CONTINUE.
 - ✓ Sử dụng câu lệnh CONTINUE để quay lại vòng lặp WHILE và thực thi các điều kiện tiếp theo.



TẠO MỘT THỦ TỤC LƯU TRỮ ĐƠN GIẢN

- **Cú pháp:**

CREATE PROC *name-proc* [(*parameters*)]

[**with RECOMPILE** | **ENCRYPTION**]

AS

BEGIN

các câu lệnh của thủ tục

END;

TẠO MỘT THỦ TỤC LƯU TRỮ ĐƠN GIẢN

```
CREATE PROCEDURE SelectCustomerstabledata  
AS  
BEGIN  
    SELECT *  
    FROM Testdb.Customers  
END;
```

Sử dụng câu lệnh EXECUTE hoặc EXEC

❑ Cú pháp:

EXECUTE sp_name;

Hoặc

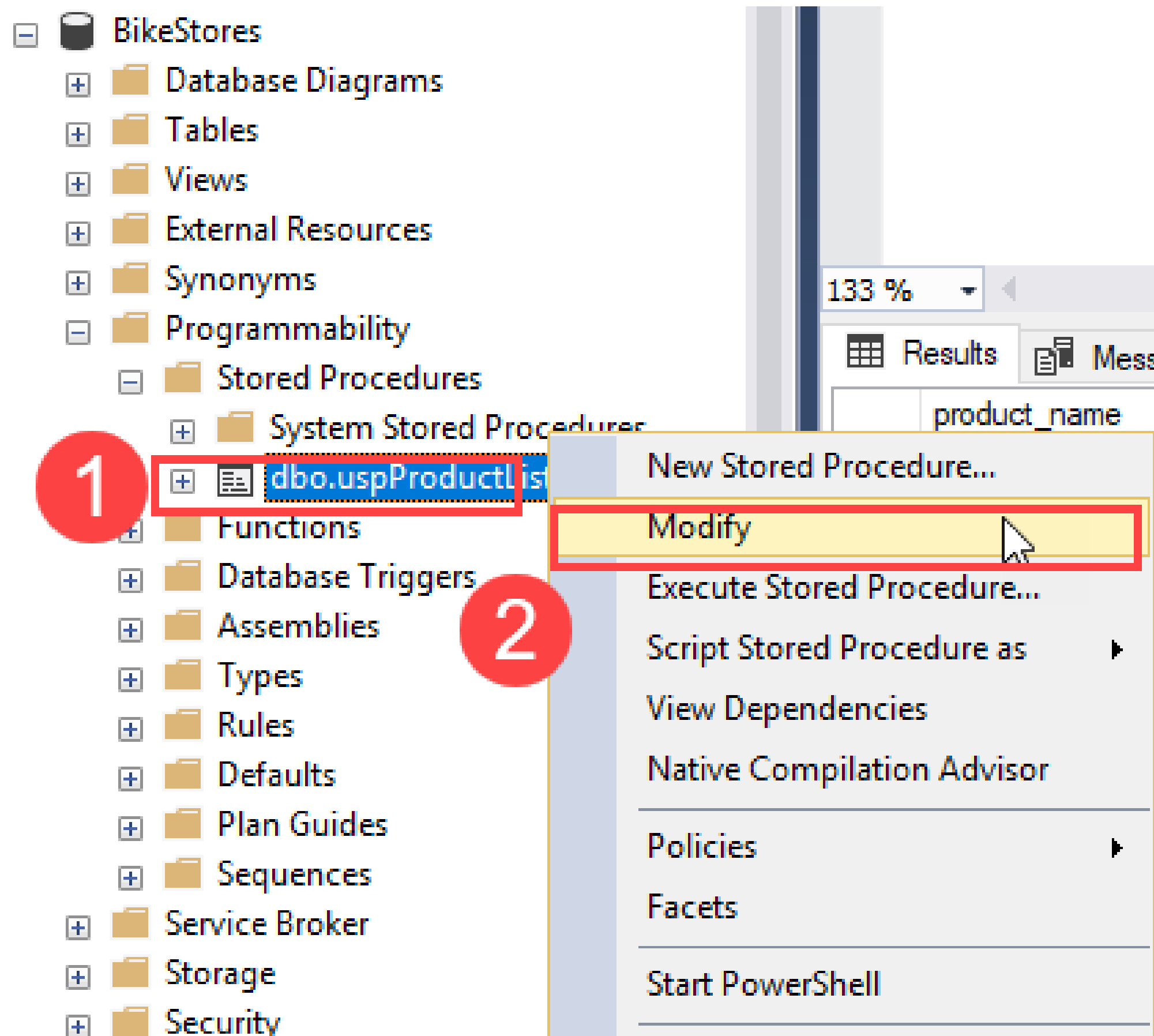
EXEC sp_name;

❑ Ví dụ:

EXEC uspProductList;

SỬA ĐỔI THỦ TỤC ĐÃ LƯU TRỮ

Bước 1:



Bước 2:

```
ALTER PROCEDURE uspProductList
AS
BEGIN
    SELECT
        product_name,
        list_price
    FROM
        production.products
    ORDER BY
        list_price
END;
```

XÓA MỘT THỦ TỤC ĐÃ LƯU TRỮ

Cú pháp:

DROP PRO sp_name;

Hoặc

DROP PROCEDURE sp_name;

SỬ DỤNG BIẾN TRONG THỦ TỤC

- Bên trong thủ tục có thể sử dụng các biến để: lưu trữ các giá trị tính toán hoặc truy xuất đến CSDL
- Các biến được khai báo với **declare** theo:
DECLARE @tên-biến *kiểu-dữ-liệu*

CÁC KIỂU TRẢ VỀ CỦA THỦ TỤC LƯU TRỮ

- Sử dụng các tham số output
- Sử dụng các lệnh trả về giá trị, luôn trả về số nguyên
- Tập các giá trị trả về của câu lệnh SQL
- Một biến con trỏ toàn cục có thể tham chiếu từ ngoài thủ tục

CÁC KIỂU TRẢ VỀ CỦA THỦ TỤC LƯU TRỮ

*Giá trị trả về cho thủ tục

- Cho phép thủ tục trả lại một giá trị nào đó sau khi thực hiện
- **Cú pháp:** thêm **RETURN** *giá-trị-trả-về*

CREATE PROC tuoicaonhat

AS

BEGIN

declare @maxtuoi int

select @maxtuoi = max(year(getdate()) - year(ngaysinh))

from sinhvien

return @maxtuoi

END

CÁC KIỂU TRẢ VỀ CỦA THỦ TỤC LƯU TRỮ

***Giá trị trả về thông qua tham số \Leftarrow tham biến**

- Các tham số truyền của thủ tục mặc định là truyền theo kiểu tham trị
- Để lưu lại thay đổi của đối số, sử dụng:

@tên-tham-số *kiểu-dữ-liệu* **OUTPUT**

Hoặc

@tên-tham-số *kiểu-dữ-liệu* **OUT**

- Khi thực hiện gọi phải thêm OUTPUT hoặc OUT

THAM SỐ VỚI GIÁ TRỊ MẶC ĐỊNH

- Là trường hợp các tham số của thủ tục được truyền giá trị mặc định => thực hiện lời gọi không cần đưa giá trị vào
- **Cú pháp:**
@tên-tham-số kiểu-dữ-liệu = *giá-trị-mặc-định*

TẠO THỦ TỤC SỬ DỤNG WILDCARD

- **Tạo thủ tục sử dụng Wildcard**
 - %: chuỗi kí tự bất kì
 - _: 1 kí tự
 - []: kí tự đơn bất kì thuộc giới hạn chỉ định (vd: [a-f] hay [abcdef])
 - [^]: kí tự đơn bất kì không nằm trong giới hạn chỉ định ([^a-f] hay [^abcdef])

- **Ví dụ:**

```
CREATE PROC ViewDSNV
```

```
    @TenNV nvarchar(30) = N'Nguyễn%'
```

```
AS
```

```
    SELECT * FROM NHANVIEN WHERE HoTen like @TenNV
```

```
EXEC ViewDSNV
```

- Cú pháp:
ALTER PROC Tên_thủ_tục [danh-sách-tham-số]
AS
BEGIN
 <Tập lệnh>
END
- Sử dụng tương tự như CREATE PROC
- Không làm thay đổi quyền được cấp trên thủ tục
- Không tác động đến các thủ tục khác hay trigger phụ thuộc thủ tục này

THỦ TỤC THÊM DỮ LIỆU VÀO BẢNG

- ☐ Là thủ tục cho phép thêm dữ liệu vào một bảng nào đó
- ☐ Khi thêm phải thỏa mãn các ràng buộc liên quan

THỦ TỤC THÊM DỮ LIỆU VÀO BẢNG

- ✓ Loại thủ tục này có tham số vào là tên các cột có trong bảng, trừ các cột có kiểu dữ liệu tự tăng (identity)

*IF Exists (Select * from Ten_Bang where
Ten_Cot=@ten_cot)*

BEGIN

print 'thông báo lỗi'

END

THỦ TỤC THÊM DỮ LIỆU VÀO BẢNG

- Kiểm tra ràng buộc khóa ngoại (Foreign Key)

- Cú pháp

*if not Exists (Select * from Ten_Bang where Ten_Cot=@ten_cot)
begin*

print 'thông báo lỗi'

set @biến_trả_về = giá trị lỗi

return

End

- Kiểm tra ràng buộc Miền giá trị (Check)

if @ten_cot không nằm trong miền giá trị

begin

print 'thông báo lỗi'

End

THỰC HÀNH



- ✓ Câu hỏi trắc nghiệm
- ✓ Bài tập

