

Hướng dẫn thực hành môn QTCSDDL

Thực hiện khai báo ràng buộc toàn vẹn miền giá trị sử dụng rule

B1: Tạo rule

```
CREATE RULE r_MACT
AS @MaCT IN ( 'CT1', 'CT2' )
```

B2: Gắn rule vào một thuộc tính của bảng. Khi thực hiện bắt buộc rule phải được tạo trước.

```
SP_BINDRULE 'r_MaCT', 'ChươngTrinh.MaCT'
```

Để thực hiện đồng thời B1 và B2. Thêm GO vào giữa.

```
CREATE RULE r_MACT
AS @MaCT in ( 'CT1', 'CT2' )
GO
SP_BINDRULE 'r_MaCT', 'ChươngTrinh.MaCT'
```

- **Gỡ bỏ rule khỏi thuộc tính của bảng.**

```
SP_UNBINDRULE 'ChươngTrinh.MaCT'
```

- **Hủy rule.**

Khi dùng bắt buộc rule phải được gỡ bỏ khỏi **tất cả** các thuộc tính của bảng mà nó đã được gắn vào.

```
DROP RULE r_MACT
```

1. **So sánh việc dùng rule và check constraint**

- o Constraint gắn đặc thù với một thuộc tính cụ thể của một bảng
- o Rule có thể áp dụng cho nhiều thuộc tính của nhiều bảng khác nhau có cùng RBTV về miền giá trị → Sử dụng Rule khi có nhiều RBTV về miền giá trị giống nhau trong CSDL.

```
CREATE RULE r_SoTietToiDa
AS @SoTiet <= 120
GO
SP_BINDRULE 'r_SoTietToiDa', 'GiangKhoa.SoTiet_LT'
GO
SP_BINDRULE 'r_SoTietToiDa', 'GiangKhoa.SoTiet_TH'
```

HƯỚNG DẪN THỰC HÀNH

Stored procedure & Triggers

Mục đích:

- Xây dựng các stored procedure và trigger để thực hiện các chức năng của hệ thống.

Hướng dẫn:

1. Stored procedure:

Khi chúng ta tạo một ứng dụng với Microsoft SQL Server, ngôn ngữ lập trình Transact-SQL là ngôn ngữ chính giao tiếp giữa ứng dụng của chúng ta và database của SQL Server. Khi chúng ta tạo các chương trình bằng Transact-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 gửi các lệnh đến SQL Server và xử lý các kết quả, hoặc chúng ta có thể lưu trữ những chương trình như các stored procedure trong SQL Server và tạo những ứng dụng để thực thi các stored procedure và xử lý các kết quả.

Stored procedure trong SQL Server cũng giống như các thủ tục trong các ngôn ngữ lập trình khác. Chúng có các đặc tính sau:

- Chấp nhận những tham số vào và trả về những giá trị được chứa trong các tham số ra để gọi những thủ tục hoặc xử lý theo lô.
- Chứa các lệnh của chương trình để thực hiện các xử lý trong database, bao gồm cả lệnh gọi các thủ tục khác thực thi.
- Trả về các trạng thái giá trị để gọi những thủ tục hoặc thực hiện các xử lý theo lô để biết việc thực hiện thành công hay thất bại, nếu thất bại thì lý do vì sao thất bại.

Chúng ta có thể dùng Transact – SQL EXECUTE để thực thi các stored procedure. Stored procedure khác với các hàm xử lý 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 Transact-SQL lưu trữ cục bộ là:

- Chúng 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.
- Chúng 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.
- Chúng 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 gửi hàng trăm lệnh đó lên mạng và điều này sẽ dẫn đến tình trạng kẹt mạng.
- Chúng 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.

Một Stored procedure được định nghĩa gồm những thành phần chính sau:

- Tên của stored procedure
- Các tham số
- Thân của stored procedure: bao gồm các lệnh của Transact-SQL dùng để thực thi procedure.

Một stored procedure được tạo bằng lệnh Create Procedure, và có thể thay đổi bằng cách dùng lệnh Alter Procedure, và có thể xóa bằng cách dùng lệnh Drop Procedure trong lập lệnh của Transact – SQL.

- Cú pháp của lệnh Create Procedure

```
CREATE PROCEDURE procedure_name
  {@parameter data_type input/output} /*các biến tham số vào ra*/
AS
Begin
  [khai báo các biến cho xử lý]
  {Các câu lệnh transact-sql}
End
```

- Cú pháp của lệnh Alter Procedure

```
ALTER PROCEDURE procedure_name
  [ {@parameter data_type } ]
AS
Begin
  [khai báo các biến cho xử lý]
  {Các câu lệnh transact-sql}
End
```

- Cú pháp của lệnh Drop Procedure

```
DROP PROCEDURE procedure_name
```

2. Trigger:

Trigger là một trường hợp đặc biệt của store procedure, nó sẽ có hiệu lực khi chúng ta thay đổi dữ liệu trên một bảng dữ liệu cụ thể, hoặc các xử lý làm thay đổi dữ liệu của các lệnh: insert, update, delete. Trigger có thể chứa các lệnh truy vấn từ các bảng khác hoặc bao gồm những lệnh SQL phức tạp.

Một số thuận lợi khi sử dụng trigger:

- Trigger chạy một cách tự động: chúng được kích hoạt ngay tức thì khi có sự thay đổi dữ liệu trên bảng dữ liệu.
- Trigger có thể thực hiện cascade khi việc thi hành có ảnh hưởng đến những bảng liên quan.
- Trigger có những hiệu lực ít bị hạn chế hơn so với ràng buộc giá trị nghĩa là có thể ràng buộc tham chiếu đến những cột của những bảng dữ liệu khác.

Cú pháp các lệnh liên quan đến trigger.

- Lệnh tạo Trigger:

```
Create Trigger trigger_name
    on table_name
    for [insert,update,delete]
As
Begin
    {Khai báo các biến xử lý}
    {Các lệnh Transact-SQL}
End
```

- Lệnh xóa Trigger:

```
Drop Trigger trigger_Name
```