



Quản trị CSDL SQL Server

#9. **Lập trình T-SQL**

Lương Trần Hy Hiến

Nội dung

- T-SQL
- Khai báo biến
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Các lỗi



Transact-SQL (T-SQL)

- T-SQL tổ chức theo từng khối lệnh, một khối lệnh có thể lồng bên trong một khối lệnh khác, một khối lệnh bắt đầu bởi BEGIN và kết thúc bởi END, bên trong khối lệnh có nhiều lệnh, và các lệnh ngăn cách nhau bởi dấu chấm phẩy.
- Cấu trúc:

```
BEGIN
```

```
-- Khai báo biến
```

```
-- Các câu lệnh T-SQL
```

```
END;
```

Khai báo biến

- Khai báo biến:
`DECLARE @Tên_Biến Kiểu_Dữ_Liệu`
- Ví dụ:
`DECLARE @Tuoi int`
`DECLARE @MSSV varchar(5)`
`DECLARE @numCount int`
- Tên biến: Bắt đầu bởi @
- Kiểu dữ liệu của biến: Lấy kiểu dữ liệu hệ thống, trừ kiểu text, ntext, image

Gán giá trị cho biến

- Cách 1:

SET @Tên_Biến = Giá_Trị

- Ví dụ:

DECLARE @HoTen nvarchar(20)

SET @HoTen = N'Nguyễn Hằng Nga'



Gán giá trị cho biến (tt)

- Cách 2:

```
SELECT @Tên_Biến = Giá_Trị
```

- Ví dụ:

```
DECLARE @HoTen nvarchar(20)
```

```
SELECT @HoTen = N'Nguyễn Hằng Nga'
```



Gán giá trị cho biến (tt)

- Cách 3:

```
SELECT @Tên_Biến = Tên_cột  
FROM Tên_Bảng
```

- Ví dụ: Tìm lương lớn nhất của tất cả nhân viên:

```
DECLARE @MaxSalary decimal(18,2)  
SELECT @MaxSalary = MAX(Luong)  
FROM NhanVien
```


Ví dụ về biến kiểu ngày tháng

```
DECLARE @date date;
```

```
SET @date = '2019-4-16';
```

```
DECLARE @curent_date date = getdate();
```

```
DECLARE @duedate date = getdate()+2;
```

```
SELECT @date AS 'Date',
```

```
    @curent_date AS 'Curent date',
```

```
    @duedate AS 'Due date';
```



Ví dụ về biến kiểu table

```
DECLARE @TableVariable table(  
    course_name varchar(200),  
    course_price int);  
  
INSERT INTO @TableVariable (course_name, course_price)  
    SELECT name, price  
    FROM Training_Course WHERE id in (1,2);  
  
SELECT course_name, course_price FROM @TableVariable;
```

Cấu trúc điều khiển

Nếu (biểu thức điều kiện) thì
Lệnh/Khối lệnh

IF (biểu thức điều kiện)

BEGIN

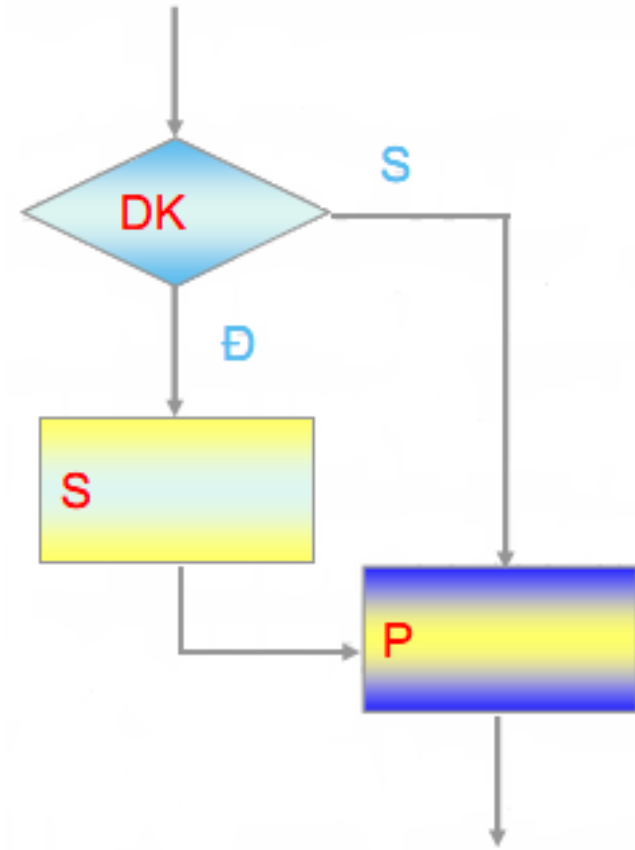
Lệnh/Khối lệnh S

hoặc SQL Statement

END

Lệnh/Khối lệnh P

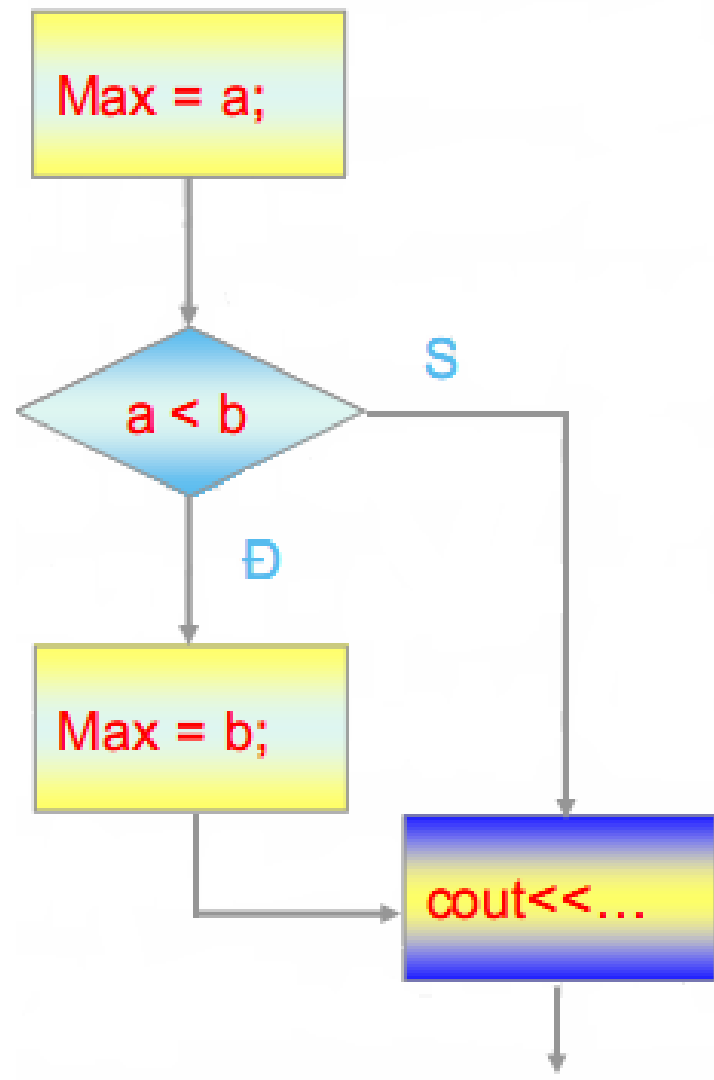
hoặc SQL Statement



Cấu trúc điều khiển

- Tìm **Max** 2 số

```
DECLARE @a, @b, @Max int
SET @Max = @a
IF (@a < @b)
BEGIN
    SET @Max = @b
END
PRINT @Max
```



Cấu trúc điều kiện

- Nếu (biểu thức điều kiện) thì
Lệnh/Khối lệnh S1
- Ngược lại
Lệnh/Khối lệnh S2

IF (biểu thức điều kiện)

BEGIN

Lệnh/Khối lệnh S1

END

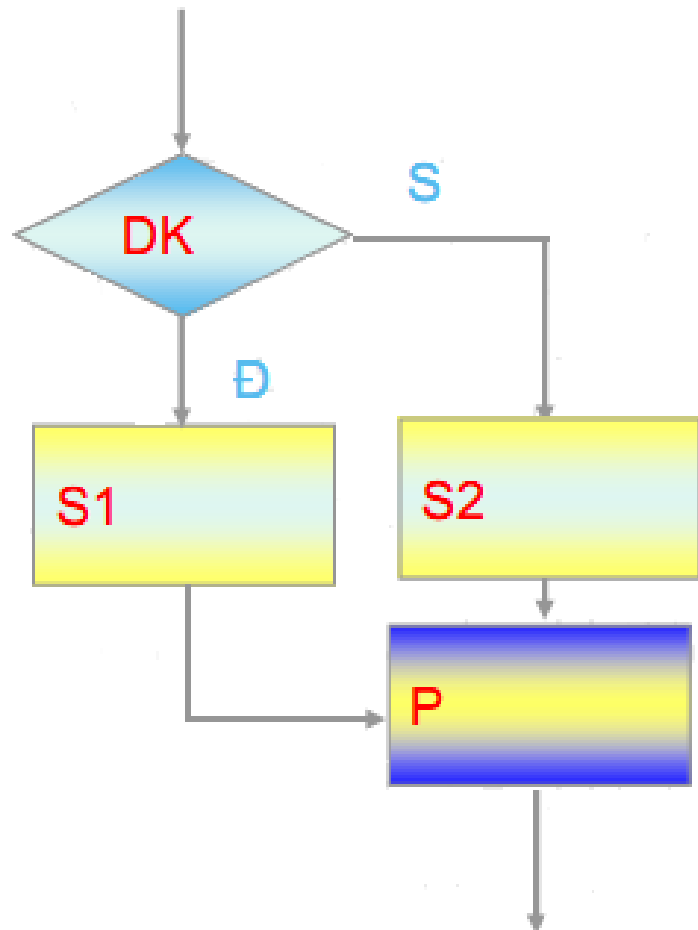
ELSE

BEGIN

Lệnh/Khối lệnh S

END

Lệnh/Khối lệnh P



Ví dụ tìm GTLN 2 số

```
DECLARE @a, @b, @Max int
IF (@a < @b)
BEGIN
    SELECT @Max = @b
END
ELSE
BEGIN
    SELECT @Max = @a
END
Print @Max
```



Cấu trúc CASE

- Cho phép kiểm tra điều kiện và xuất thông tin theo từng trường hợp
- Cú pháp 1

CASE <tên cột>/<biểu thức>

WHEN <giá trị> **THEN** <biểu thức>

WHEN <giá trị> **THEN** <biểu thức>

...

[**ELSE** <biểu thức>]

END

Cấu trúc CASE

- Cú pháp 2

CASE

WHEN <giá trị> **THEN** <biểu thức>

WHEN <giá trị> **THEN** <biểu thức>

...

[**ELSE** <biểu thức>]

END

Ví dụ: Giảm 5% giá bán nếu khách hàng là Nữ,
2% giá bán nếu khách hàng là Nam

Cấu trúc lặp

```
WHILE (biểu thức logic)
```

```
BEGIN
```

```
--Lệnh/Khoi lệnh
```

```
END
```

- Viết chương trình tính tổng
 $s = 1 + 2 + \dots + n$

```
DECLARE @i INT
```

```
DECLARE @N INT
```

```
DECLARE @S INT
```

```
SELECT @i=1, @s=0
```

```
WHILE (@i <= @N)
```

```
BEGIN
```

```
    SELECT @S = @S + @i
```

```
    SELECT @i = @i + 1
```

```
END
```

```
PRINT @S
```



Cấu trúc LẶP

- **BREAK**: Thoát khỏi vòng lặp WHILE
- **CONTINUE**: Thực hiện lần lặp mới



Một số biến toán cục

- **@@ERROR**: giá trị lỗi về trong biến toàn cục
 - @@ERROR= 0: không xảy ra lỗi
 - @@ERROR <> 0: xảy ra lỗi với mã lỗi là @@ERROR
- **@@IDENTITY**: Giá trị tự tăng gần nhất
- **@@ROWCOUNT**: Số lượng các hàng bị tác động bởi câu lệnh SQL gần nhất.

Một số biến toán cục (tt)

Danh sách các biến	Mục đích
@@CONNECTIONS	Số lượng các kết nối tới server kể từ lần khởi động cuối cùng.
@@CPU_BUSY	Số lượng một phần nghìn giây của hệ thống đang xử lý kể từ khi SQL Server được khởi động.
@@CURSOR_ROWS	Số hàng trong thời gian truy cập gần nhất.
@@DATEFIRST	The current value of SET DATEFIRST parameter which controls the first day that's considered to be the first day of the week.
@@FETCH_STATUS	0 if the last fetch status was successful. -1 if there was an error
@@LANGUAGE	Name of the language currently in use.
@@MAX_CONNECTIONS	Số lượng các kết nối hiện tại tối đa có thể tạo được.
@@SERVERNAME	Tên của máy chủ cục bộ
@@SERVICENAME	Tên của dịch vụ SQL trên máy tính
@@TIMETICKS	Số lượng phần triệu giây trên mỗi giây tại máy tính hiện thời
@@TRANSCOUNT	Số lượng các giao dịch mở trên kết nối hiện tại
@@VERSION	Thông tin về phiên bản của SQL Server

Điều khiển ngoại lệ

- Sử dụng BEGIN TRY .. END TRY để bắt lỗi.
- Nếu lỗi xảy ra trong khối này
- nó sẽ nhảy vào khối BEGIN CATCH .. END CATCH.

BEGIN TRY

Lệnh xử lý

END TRY

- BEGIN CATCH .. END CATCH phải được đặt ngay

- phía sau của khối BEGIN TRY .. END TRY.

BEGIN CATCH

Xử lý lỗi

END CATCH

Thông tin lỗi

Hàm	Mô tả
ERROR_NUMBER()	Trả về mã lỗi.
ERROR_MESSAGE()	Trả về văn bản đầy đủ của các thông báo lỗi. Các văn bản bao gồm các giá trị đã cung cấp cho các tham số, chẳng hạn như độ dài, tên đối tượng, hoặc thời gian.
ERROR_SEVERITY()	Trả về mức độ nghiêm trọng của lỗi.
ERROR_STATE()	Trả về trạng thái lỗi
ERROR_LINE()	Trả về số vị trí dòng code gây lỗi.
ERROR_PROCEDURE()	Trả về tên của stored procedure hoặc trigger nơi mà lỗi phát ra.

Tham khảo

- <https://www.tsql.info>
- <https://www.sqlservertutorial.net>



Q&A

HEALTH