



TRƯỜNG ĐẠI HỌC VINH

CHƯƠNG 2

LẬP TRÌNH CƠ BẢN VỚI T-SQL

Phan Anh Phong PhD.
Viện Kỹ thuật và Công nghệ

1

Nội dung



- **Biến và các phép toán**
- Cấu trúc điều khiển
- Làm việc với cursor
- Bài tập
- Hướng dẫn tự học

2

Biến



- Biến là 1 vùng nhớ dùng để lưu trữ dữ liệu, được xác định qua tên biến và kiểu dữ liệu
- Trong T-SQL có 2 loại biến:
 - Hệ thống:
 - Cục bộ
- Biến hệ thống do T-SQL định nghĩa sẵn và bắt đầu bằng ký hiệu @@.
- Ví dụ:
 - Không thay đổi tên, kiểu của biến hệ thống, người dùng có thể sử dụng biến hệ thống
- **Biến cục bộ do người dùng tự định nghĩa và PHẢI KHAI BÁO bắt đầu bằng @**
 - Ví dụ: `DECLARE @ten_lop VARCHAR(50);`

3

Ví dụ biến hệ thống



- **Một số biến hệ thống thường dùng**
 - @@error - Mã lỗi trả về của lệnh T-SQL
 - @@rowcount – số dòng bị ảnh hưởng bởi câu lệnh T-SQL
 - @@SERVERNAME -trả về tên của máy chủ CSDL
 - @@fetch_status – sử dụng với cursor, đọc dữ liệu trong bảng theo từng dòng cursor. Khi đọc mẫu tin thành công thì biến có giá trị =0

4

Khai báo biến



- -- Khai báo một biến
- Declare @Result Int
- -- Khai báo một biến có giá trị 50
- Declare @a Int = 50
- -- Khai báo một biến có giá trị 2020
- Declare @b Int = 2020
- -- In ra màn hình Console
- -- Sử dụng Cast để ép kiểu Int về kiểu chuỗi ký tự.
- -- Sử dụng toán tử + để nối 2 chuỗi ký tự
- Print 'a= ' + Cast(@a as varchar(15))
- -- In ra màn hình Console
- Print 'b= ' + Cast(@b as varchar(15))
- -- Tính tổng
- Set @Result = @a + @b
- -- In ra màn hình Console
- Print 'Result= ' + Cast(@Result as varchar(15))

5

Biến cục bộ



- Biến cục bộ có giá trị trong một lô (*query batch*) hoặc trong một stored procedure/ function

```
DECLARE @cb_id int  
SET @CB_ID=1  
GO
```

```
PRINT @CB_id
```

BỊ LỖI, vì sao?

Giải thích ở phần sau

6

Gán giá trị cho biến



- Lệnh SET @<ten_bien> = <Biểu_thức>
- Lệnh SELECT @<ten_bien> = <Biểu_thức>

7

Ví dụ phép gán



- **Giả sử có bảng:**
CREATE TABLE canbo
(macb int primary key, ten varchar(15), namsinh int)
Với nội dung:
INSERT INTO canbo VALUES
(1,'May', 1990),
(2,'Hoc', 1991)
- **Gán bằng lệnh SET:**
USE HQTCSDL59
DECLARE @soluongcb int
SET @soluongcb=COUNT(*) FROM canbo
PRINT @soluongcb
LỖ!!!

8

Phan Anh Phong – Vinh University

Ví dụ phép gán



```
USE HQTCSDL59
DECLARE @soluongcb int
SET @soluongcb=COUNT(*) FROM canbo
PRINT @soluongcb
LỖI!!
SỬA LẠI:
USE HQTCSDL59
DECLARE @soluongcb int
SET @soluongcb=(SELECT COUNT(*) FROM canbo)
PRINT @soluongcb
GIẢI THÍCH?
```

9

Phan Anh Phong – Vinh University

Ví dụ phép gán



- Gán bằng lệnh **SELECT**:

```
USE HQTCSDL59
DECLARE @soluongcb int
SELECT @soluongcb=COUNT(*) FROM canbo
PRINT @soluongcb
```

10

Phan Anh Phong – Vinh University

Ví dụ phép gán



- **Gán bằng lệnh SELECT:**

```
DECLARE @bien1 int, @bien2 varchar(15), @bien3 int
SELECT @bien1 = macb, @bien2 = ten, @bien3 = namsinh
FROM canbo
PRINT @bien1
PRINT @bien2
PRINT @bien3
```

- Lưu ý: nếu câu truy vấn trả về nhiều dòng, các biến chỉ nhận giá trị tương ứng của dòng cuối trong bảng
- Lấy giá trị dòng đầu: SELECT TOP 1 ... FROM ...

11

Phan Anh Phong – Vinh University

Khối lệnh



```
BEGIN
-- khai báo biến
-- các lệnh T-SQL
END
```

12

Xử lý lô (Batch)



Một chương trình viết bằng T-SQL có thể chia thành nhiều lô (batch) để xử lý.

Mỗi lô là một (một nhóm) các lệnh T-SQL được phân cách nhau bằng lệnh GO.

Mình lệnh GO viết trên một dòng, ngược lại bị LỖI

Ví dụ

```
SELECT * FROM canbo; GO
```

– LỖI

Msg 102, Level 15, State 1, Line 1
Incorrect syntax near 'GO'.

SỬA LẠI:

```
SELECT * FROM canbo  
GO
```

Lưu ý: Lệnh GO là lệnh của SSMS chứ không phải lệnh của T-SQL, do vậy nếu dung GO ở môi trường khác SSMS sẽ bị lỗi

13

Xử lý lô (Batch)



```
DECLARE @cb_id int
```

```
SET @CB_ID=1
```

```
GO
```

```
PRINT @CB_id
```

BỊ LỖI, vì sao?

Chương trình trên có mấy lô?

Biến @cb_id là biến cục bộ hay toàn cục?

→ CÁCH SỬA

```
DECLARE @cb_id int
```

```
SET @CB_ID=1
```

```
PRINT @CB_id
```

```
GO
```

14

Nội dung



- Biến và các phép toán
- Cấu trúc điều khiển
- Làm việc với cursor
- Bài tập
- Hướng dẫn tự học

15

IF ... else...



- Dựa vào điều kiện để quyết định những lệnh T-SQL nào sẽ được thực hiện
- Cú pháp:
If <biểu_thức_điều_kiện>
 <Lệnh| Khối_lệnh >
[Else Lệnh| Khối_lệnh]
- Khối lệnh là một hoặc nhiều lệnh nằm trong cặp từ khóa begin...end

16

IF ... else...



- Ví dụ cho CSDL có 2 bảng

HocPhan(MaHP char(5), TenHP varchar (30), SiSo int)

DangKyHoc(MaSV char(10), MaHP char(5))

Viết chương trình để thêm một đăng ký mới cho sinh viên có mã số 2020 vào lớp học phần HP01 (giả sử học phần này đã tồn tại trong bảng HocPhan). Qui định rằng mỗi học phần chỉ được đăng ký tối đa 30 sinh viên.

17

IF ... else...



- Ví dụ

Declare @SiSo int

select @SiSo = SiSo from HocPhan where MaHP="HP01"

if @SiSo < 30

 Begin

 insert into DangKyHoc(MaSV, MaHP)
 values("2020", "HP01")

 print N"Đăng ký thành công"

 End

Else

 print N "Học phần đã đủ số lượng"

18