



# TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

## Khoa Khoa học và Kỹ thuật Thông tin

Thực hành - IE103.N11.LT.1

### Lab 3.1

# MỘT SỐ CÂU LỆNH LẬP TRÌNH CƠ BẢN TRONG SQL

# Nội dung

**Comment**

**Câu lệnh PRINT**

**Câu lệnh DECLARE**

**Cấu trúc lựa chọn IF... ELSE...**

**Cấu trúc lặp WHILE...**



# Comment

19/10/2022

IE103.N11.LT.1 - Quản lý thông tin

# Comment

- ❑ **Comment** (Chú thích) được sử dụng để giải thích các phần của câu lệnh SQL hoặc để ngăn việc thực thi câu lệnh SQL nằm trong ký hiệu comment.
- ❑ Những nội dung chú thích sẽ không được thực thi trong quá trình chạy lệnh SQL.

# Các dạng comment

## ❑ Single-line comment (Comment một dòng)

- ▶ Để comment một dòng, chúng ta thêm vào đầu dòng ký hiệu `--`.
- ▶ Những nội dung nằm sau ký hiệu `--` cho đến cuối dòng sẽ bị bỏ qua (không được thực thi).

### ▶ Ví dụ

- Ví dụ 1. Comment để chú thích nội dung giải thích cho câu lệnh `SELECT`.

`--Select all:`

`SELECT * FROM Customers`

# Các dạng comment

- Ví dụ 2. Comment để bỏ qua phần cuối của câu lệnh **SELECT**.

```
SELECT * FROM Customers  
--WHERE City='Berlin'
```

- Ví dụ 3. Comment để bỏ qua một câu lệnh **SELECT**.

```
--SELECT * FROM Customers  
SELECT * FROM Products
```

# Các dạng comment

## ❑ Multi-line comment (Comment nhiều dòng)

- ▶ Để comment nhiều dòng nằm gần nhau, chúng ta sử dụng cặp ký hiệu `/*` (đặt ở vị trí bắt đầu) và `*/` (đặt ở vị trí kết thúc).
- ▶ Những nội dung nằm giữa cặp ký hiệu `/*` và `*/` sẽ bị bỏ qua (không được thực thi).

# Các dạng comment

- Ví dụ 2. Comment nhiều dòng để bỏ qua nhiều câu lệnh **SELECT**.

```
/*SELECT * FROM Customers  
SELECT * FROM Products  
SELECT * FROM Orders  
SELECT * FROM Categories*/  
SELECT * FROM Suppliers
```



# Các dạng comment

## ❑ Partial comment (Comment một phần)

► Để comment một phần của dòng hoặc những phần nằm trên các dòng liên tiếp nhau, chúng ta cũng sử dụng cặp ký hiệu `/*` và `*/`.

### ► Ví dụ

- Ví dụ 1. Comment một thuộc tính được liệt kê trong câu lệnh `SELECT`.

```
SELECT CustomerName, /*City,*/ Country FROM Customers
```

# Comment

## ❑ Tổ hợp phím tắt comment trong SQL Server

(Dùng để comment nhanh nhiều câu lệnh)

- ▶ Để comment, chọn các câu lệnh cần comment, bấm tổ hợp phím **Ctrl + K + C** (Nhấn giữ phím **Ctrl**).
- ▶ Để bỏ comment, chọn các câu lệnh đã được comment, bấm tổ hợp phím **Ctrl + K + U** (Nhấn giữ phím **Ctrl**).

# Câu lệnh PRINT

19/10/2022

IE103.N11.LT.1 - Quản lý thông tin

# Câu lệnh PRINT

- Lệnh **PRINT** được sử dụng để in nội dung (chuỗi ký tự, biến cục bộ, biểu thức,...).

## ► Ví dụ

- Ví dụ 1. In chuỗi 'Hello world'.

```
PRINT 'Hello world'
```

- Ví dụ 2. In số 1.

```
PRINT 1
```

- Ví dụ 3. In kết quả của phép toán 1+1.

```
PRINT 1+1
```



# Câu lệnh PRINT

- Ví dụ 4. In kết quả của phép cộng chuỗi 'Hello' + ' ' + 'world'.

```
PRINT 'Hello' + ' ' + 'world'
```

- Ví dụ 5. In thời gian hiện tại bằng cách sử dụng hàm GETDATE().

```
PRINT GETDATE()
```

- Ví dụ 6. In chuỗi 'Count' và số 1 bằng cách sử dụng hàm CAST() để chuyển đổi kiểu dữ liệu số 1 thành kiểu VARCHAR(5).

```
PRINT 'Count' + CAST(1 AS VARCHAR(5))
```

- Ví dụ 7. In các chuỗi và số liên tiếp nhau: 'Count ', 1, ' AND ', 1 bằng cách sử dụng hàm CONCAT() để nối các chuỗi và số với nhau.

```
PRINT CONCAT('Count ', 1, ' AND ', 1)
```

# Câu lệnh PRINT

- ❑ Để in chuỗi ký tự có chứa tiếng Việt có dấu, nên thêm ký tự **N** vào trước chuỗi ký tự.

## ► Ví dụ

```
PRINT N'Xin chào Việt Nam'
```

# Câu lệnh DECLARE

19/10/2022

IE103.N11.LT.1 - Quản lý thông tin

# Câu lệnh DECLARE

- ❑ Dùng để khai báo biến cục bộ.
- ❑ Khai báo một biến (với giá trị khởi tạo):

```
DECLARE @Tên_biến Kiểu_dữ_liệu [ = Giá_trị ]
```



# Câu lệnh DECLARE

## ❑ Ví dụ

- ▶ Ví dụ 1. Khai báo biến I có kiểu dữ liệu **INT** với giá trị khởi tạo là 100.

```
DECLARE @I INT = 100
```

```
PRINT @I
```

- ▶ Ví dụ 2. Khai báo biến TestVariable có kiểu dữ liệu **VARCHAR(100)** với giá trị khởi tạo là 'Save Our Planet'.

```
DECLARE @TestVariable VARCHAR(100) = 'Save Our Planet'
```

```
PRINT @TestVariable
```

# Câu lệnh DECLARE

- ❑ Khai báo nhiều biến (với giá trị khởi tạo):

```
DECLARE @Tên_biến_1 Kiểu_dữ_liệu_biến_1 [ = Giá_trị_biến_1 ],  
        @Tên_biến_2 Kiểu_dữ_liệu_biến_2 [ = Giá_trị_biến_2 ],  
        ...  
        @Tên_biến_n Kiểu_dữ_liệu_biến_n [ = Giá_trị_biến_n ]
```

# Câu lệnh DECLARE

## ❑ Ví dụ

- ▶ Ví dụ 1. Khai báo biến var1 và biến var2 có kiểu dữ liệu **INT** với giá trị khởi tạo lần lượt là 3 và 5.

```
DECLARE @var1 INT = 3, @var2 INT = 5  
PRINT 'Var1 = ' + CAST(@var1 AS VARCHAR(5))  
PRINT 'var2 = ' + CAST(@var2 AS VARCHAR(5))  
(Sử dụng hàm CAST() để chuyển đổi kiểu dữ liệu)
```

# Câu lệnh DECLARE

- ▶ Ví dụ 2. Khai báo biến phrase1 và biến phrase2 có kiểu dữ liệu NVARCHAR(10) với giá trị khởi tạo lần lượt là 'Xin chào' và 'Việt Nam'.

```
DECLARE @phrase1 NVARCHAR(10) = N'Xin chào',  
        @phrase2 NVARCHAR(10) = N'Việt Nam'  
PRINT @phrase1 + ' ' + @phrase2
```



# Câu lệnh DECLARE

- Ngoài cách khai báo biến đi kèm giá trị khởi tạo, chúng ta có thể khai báo biến trước, sau đó gán giá trị cho biến bằng cách:

► Sử dụng lệnh **SET**:

```
DECLARE @Tên_biến Kiểu_dữ_liệu
```

```
SET @Tên_biến = Giá_trị
```

**Ví dụ.** Khai báo biến **I** có kiểu dữ liệu **INT**, sau đó gán giá trị 25.

```
DECLARE @I INT
```

```
SET @I = 25
```

```
PRINT @I
```

# Cấu trúc lựa chọn IF... ELSE...

# Cấu trúc lựa chọn IF... ELSE...

- Được sử dụng để thể hiện sự lựa chọn.
- Cú pháp:

IF Điều\_kiện

BEGIN

Các\_câu\_lệnh\_SQL

END

ELSE

BEGIN

Các\_câu\_lệnh\_SQL

END

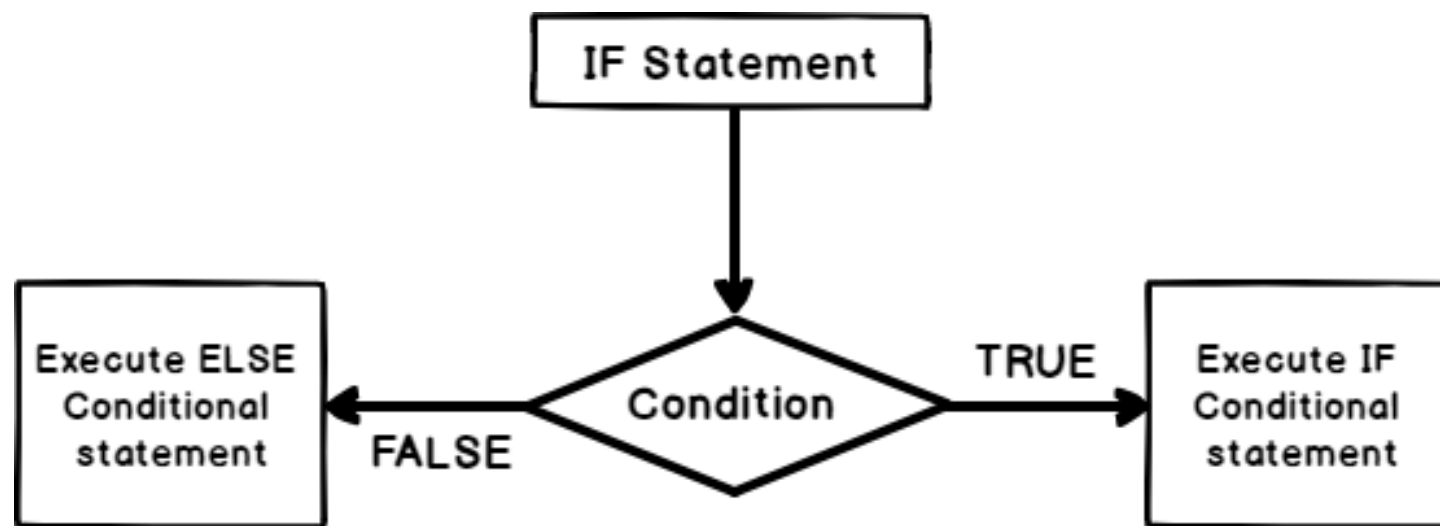
Thực hiện  
nếu điều kiện **đúng**

Thực hiện  
nếu điều kiện **sai**

# Cấu trúc lựa chọn IF... ELSE...

- ▶ Nếu điều kiện **đúng** → Thực hiện đoạn lệnh trong khối **IF**.
- ▶ Nếu điều kiện **sai** → Thực hiện đoạn lệnh trong khối **ELSE**.

□ Sơ đồ khối:





# Cấu trúc lựa chọn IF... ELSE...

## □ Lưu ý

- ▶ Từ điều kiện kiểm tra, cần nắm được:
  - Khi nào thực hiện khối lệnh **IF** . . . (khi nào điều kiện đúng ?) ?
  - Khi nào thực hiện khối lệnh **ELSE** . . . (khi nào điều kiện sai ?) ?
- ▶ Nếu khối lệnh bên trong chỉ gồm 1 câu lệnh, có thể bỏ **BEGIN** . . . **END**.

# Cấu trúc lựa chọn IF... ELSE...

## ❑ Ví dụ

- ▶ Ví dụ 1. Biểu thức điều kiện không chứa biến.

```
IF 1 = 1
```

```
→ PRINT 'Executed the statement as condition is TRUE'
```

```
ELSE
```

```
    PRINT 'Executed the statement as condition is FALSE'
```

- ▶ Ví dụ 2. Biểu thức điều kiện không chứa biến.

```
IF 2 <= 0
```

```
    PRINT 'Executed the statement as condition is TRUE'
```

```
ELSE
```

```
→ PRINT 'Executed the statement as condition is FALSE'
```

# Cấu trúc lựa chọn IF... ELSE...

- ▶ Ví dụ 3. Biểu thức điều kiện chứa biến.

```
DECLARE @StudentMarks INT = 91
```

```
IF @StudentMarks >= 80  
→ PRINT 'Passed! Congratulations!'  
ELSE  
    PRINT 'Failed! Try again!'
```

```
IF @StudentMarks < 90  
    PRINT 'Passed! Congratulations!'  
ELSE  
→ PRINT 'Failed! Try again!'
```

# Cấu trúc lựa chọn IF... ELSE...

## ❑ Lưu ý

- ▶ ELSE ... sẽ đi theo cặp với IF ... gần nhất...
- ▶ Ví dụ 4. Câu lệnh lồng.

```
DECLARE @StudentMarks INT = 91
IF @StudentMarks >= 80
    IF @StudentMarks >= 90
        PRINT 'Excellent!'
    ELSE
        PRINT 'Passed!'
ELSE
    PRINT 'Failed!'
```

# Cấu trúc lựa chọn IF... ELSE...

## □ Lưu ý

- ▶ Có thể bỏ đoạn lệnh **ELSE**..., khi đó cấu trúc lựa chọn chỉ còn **IF**...

**IF** Điều\_kiện

**BEGIN**

Các\_câu\_lệnh\_**SQL**

**END**

→ Khi điều kiện sai thì sẽ không thực hiện gì cả.



# Cấu trúc lựa chọn IF... ELSE...

- ▶ Ví dụ 5. Cấu trúc lựa chọn có IF... nhưng không có ELSE...

```
DECLARE @StudentMarks INT = 95
```

```
IF @StudentMarks >= 90
```

```
    → PRINT 'Congratulations! You are in Merit list!'
```

- ▶ Ví dụ 6. Cấu trúc lựa chọn có IF... nhưng không có ELSE...

```
DECLARE @StudentMarks INT = 85
```

```
IF @StudentMarks >= 90
```

```
    PRINT 'Congratulations! You are in Merit list!'
```

# Cấu trúc lựa chọn IF... ELSE...

## □ Cấu trúc IF... ELSE... mở rộng (IF... ELSE IF... ELSE...)

► Cú pháp:

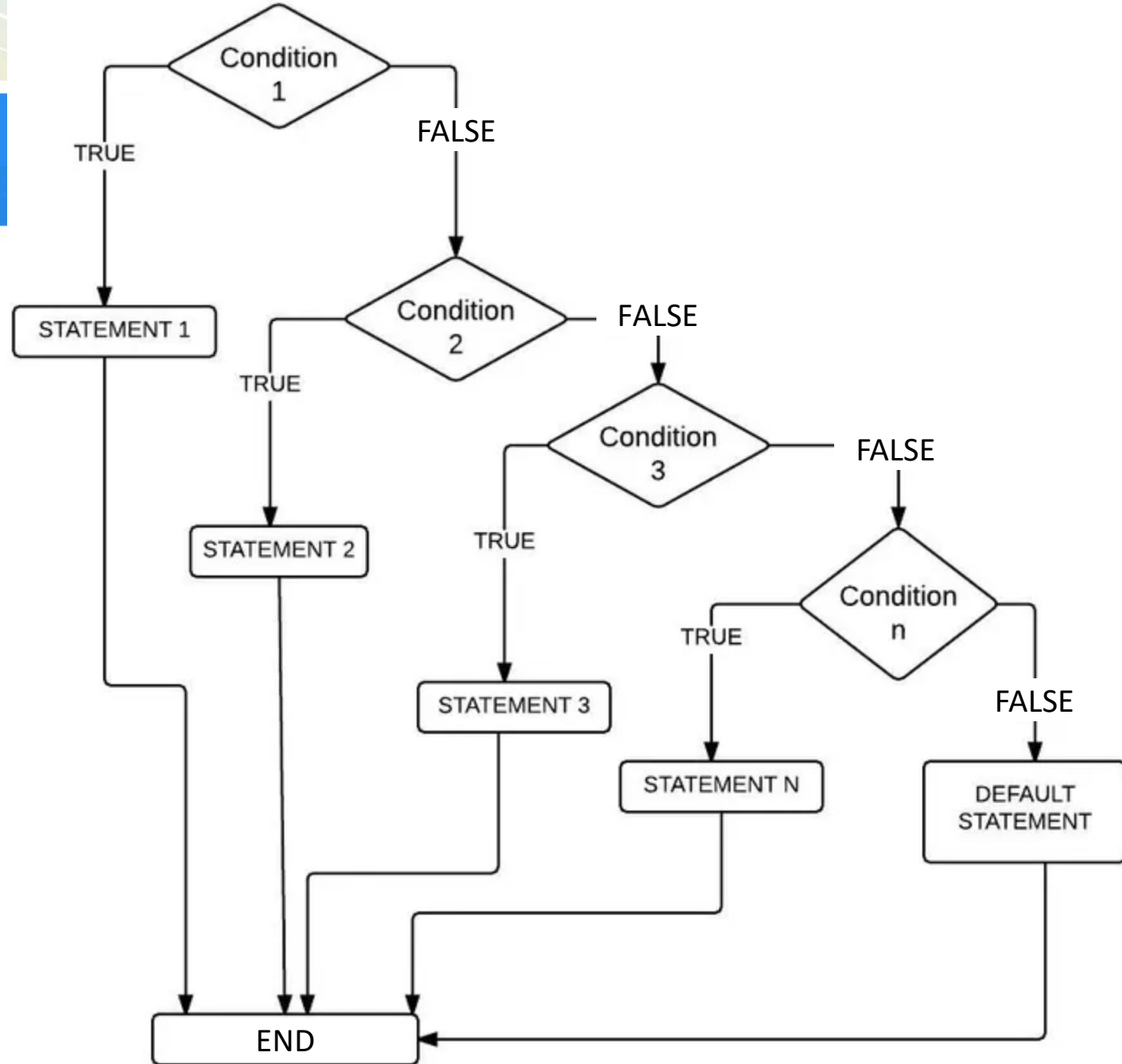
```
IF Điều_kiện_1
    BEGIN
        Các_câu_lệnh_SQL
    END
ELSE IF Điều_kiện_2
    BEGIN
        Các_câu_lệnh_SQL
    END
```

...

```
ELSE ← Khác với các điều kiện 1, 2,...
    BEGIN
        Các_câu_lệnh_SQL
    END
```

# Cấu trúc lựa chọn IF... ELSE...

► Sơ đồ khối:



# Cấu trúc lựa chọn IF... ELSE...

► Ví dụ:

```
DECLARE @A INT = 0
```

```
IF @A > 0
```

```
    PRINT 'A > 0'
```

```
ELSE IF @A < 0
```

```
    PRINT 'A < 0'
```

```
ELSE ← @A = 0
```

```
    PRINT 'A = 0'
```

# Cấu trúc lặp WHILE...

19/10/2022

IE103.N11.LT.1 - Quản lý thông tin



# Cấu trúc lặp WHILE...

- ❑ Được sử dụng để lặp lại việc thực thi câu lệnh nhiều lần.

- ❑ Cú pháp:

**WHILE** Điều\_kiện\_lặp

**BEGIN**

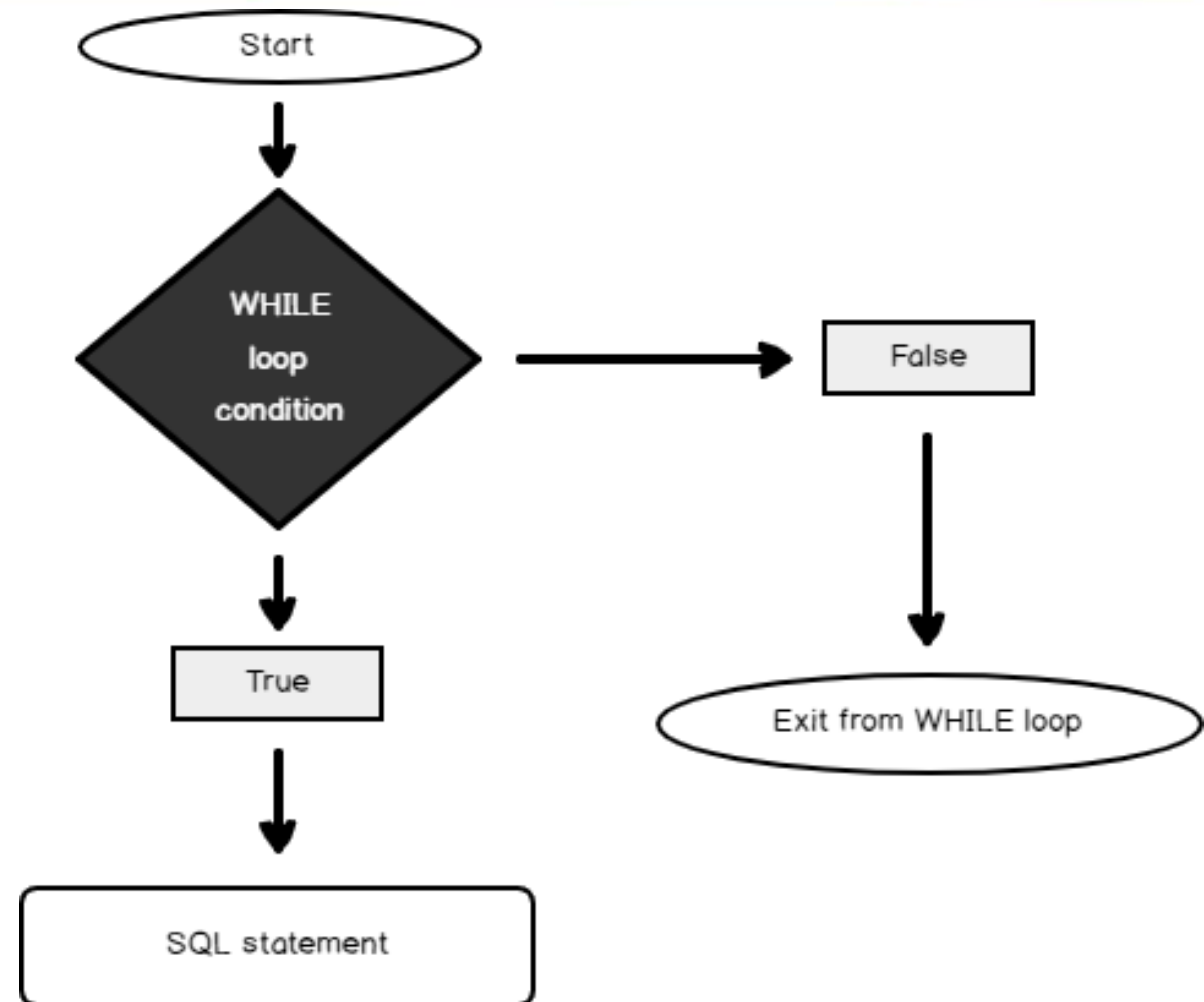
Các\_câu\_lệnh\_SQL

**END**

- ▶ Khi điều kiện lặp đúng (điều kiện lặp được thỏa mãn) thì sẽ thực hiện các câu lệnh bên trong.
- ▶ Vòng lặp sẽ dừng lại khi điều kiện lặp bị sai.

# Cấu trúc lặp WHILE...

□ Sơ đồ khối:



# Cấu trúc lặp WHILE...

- **Ví dụ.** Vòng lặp theo biến lặp @Counter với giá trị khởi tạo là 1 và điều kiện lặp @Counter <= 10, đồng thời in giá trị của biến @Counter và tăng giá trị của biến @Counter lên 1 đơn vị.

```
DECLARE @Counter INT
```

```
SET @Counter = 1
```

```
WHILE @Counter <= 10
```

```
BEGIN
```

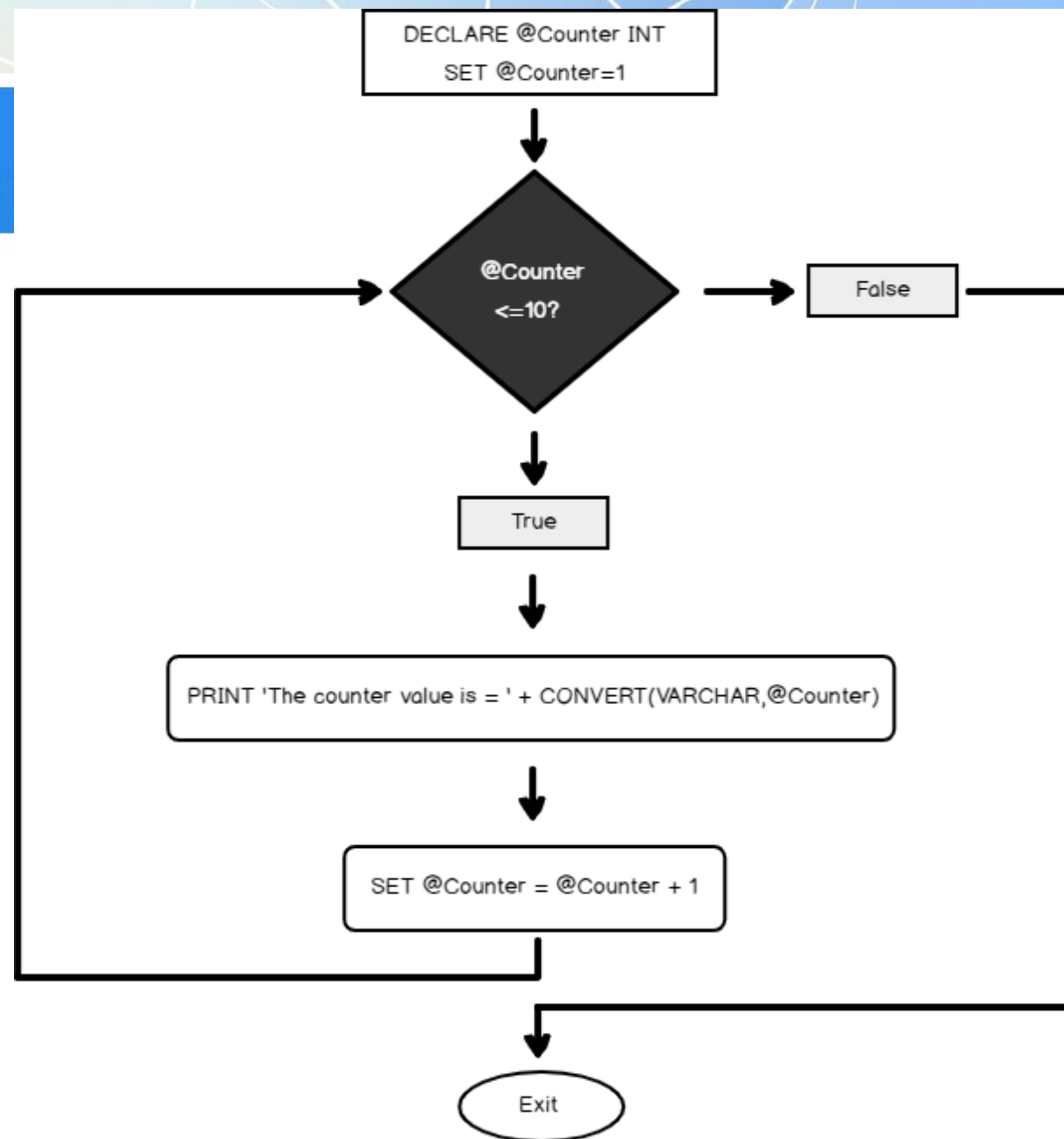
```
    PRINT 'The counter value is ' + CONVERT(VARCHAR, @Counter)
```

```
    SET @Counter = @Counter + 1
```

```
END
```

# Cấu trúc lặp WHILE...

► Sơ đồ khối của vòng lặp:



# Cấu trúc lặp WHILE...

## ► Trong ví dụ trên:

- Biến lặp:
- Giá trị khởi tạo:
- Điều kiện lặp:
- Công việc thực hiện:
- Bước thay đổi giá trị biến lặp:
- Khi nào dừng lại ?

@Counter

1

@Counter <= 10

In giá trị của biến @Counter và tăng giá trị của biến @Counter lên 1 đơn vị.

@Counter += 1

@Counter > 10



# Cấu trúc lặp WHILE...

## ❑ Lưu ý

- ▶ Cần chú ý đến điều kiện dừng của vòng lặp (tính dừng của vòng lặp), tránh để xảy ra trường hợp vòng lặp vô tận.  
(Khi nào vòng lặp dừng lại ? → Khi không thỏa điều kiện lặp nữa)
- ▶ Nếu khối lệnh bên trong chỉ gồm 1 câu lệnh, có thể bỏ **BEGIN** . . . **END**.

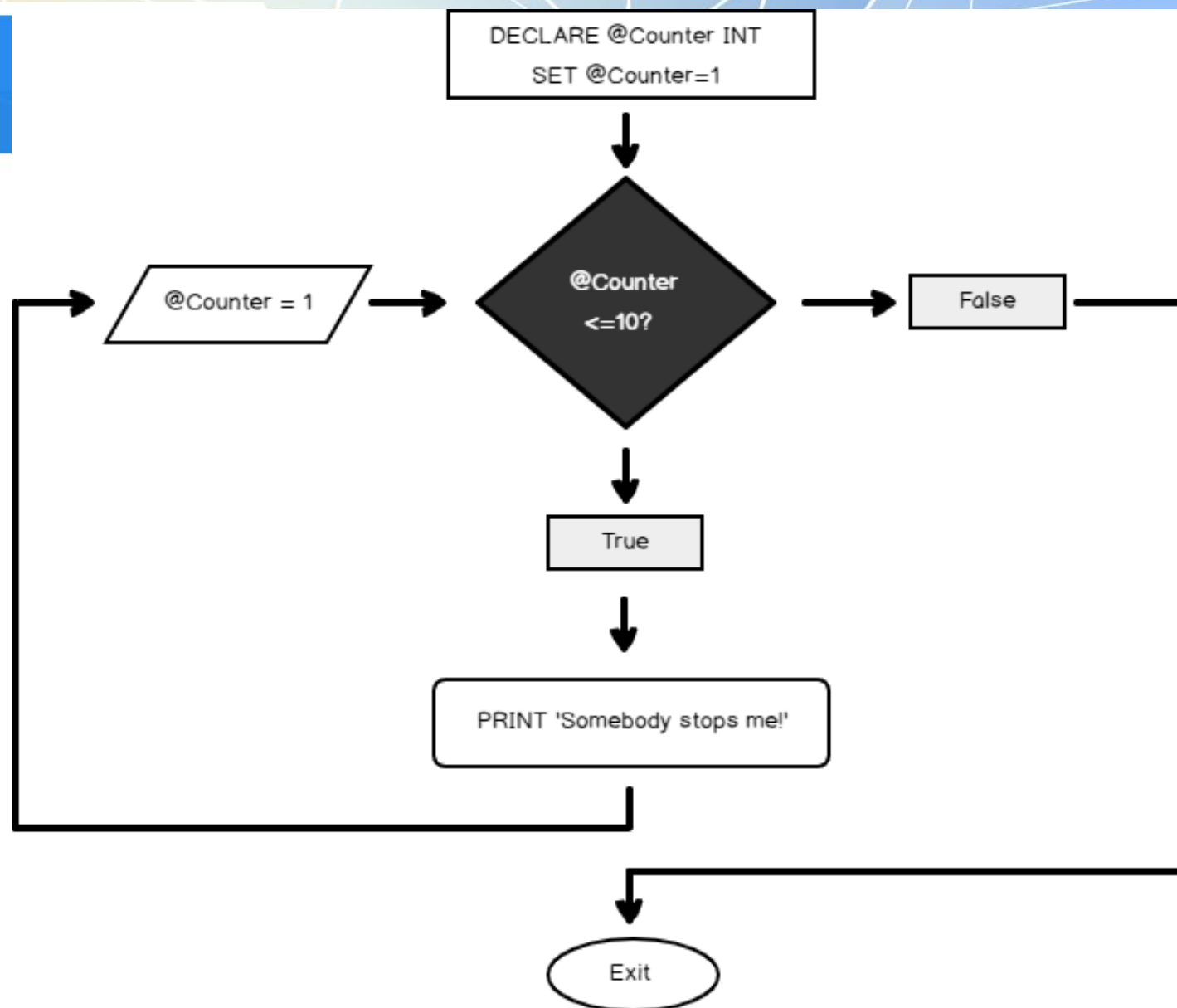
# Cấu trúc lặp WHILE...

- **Ví dụ.** Vòng lặp vô tận (Điều kiện lặp luôn luôn thỏa → Vòng lặp không bao giờ dừng lại).

```
DECLARE @Counter INT  
SET @Counter = 1  
WHILE @Counter <= 10  
    PRINT 'Somebody stops me!'
```

# Cấu trúc lặp WHILE...

► Sơ đồ khối của vòng lặp:





# CHÚC CÁC BẠN HỌC TỐT !

19/10/2022

IE103.N11.LT.1 - Quản lý thông tin