



CHƯƠNG 5

HÀM NGƯỜI DÙNG ĐỊNH NGHĨA



Con trỏ

Hàm người dùng định nghĩa

Hàm trả về giá trị vô hướng

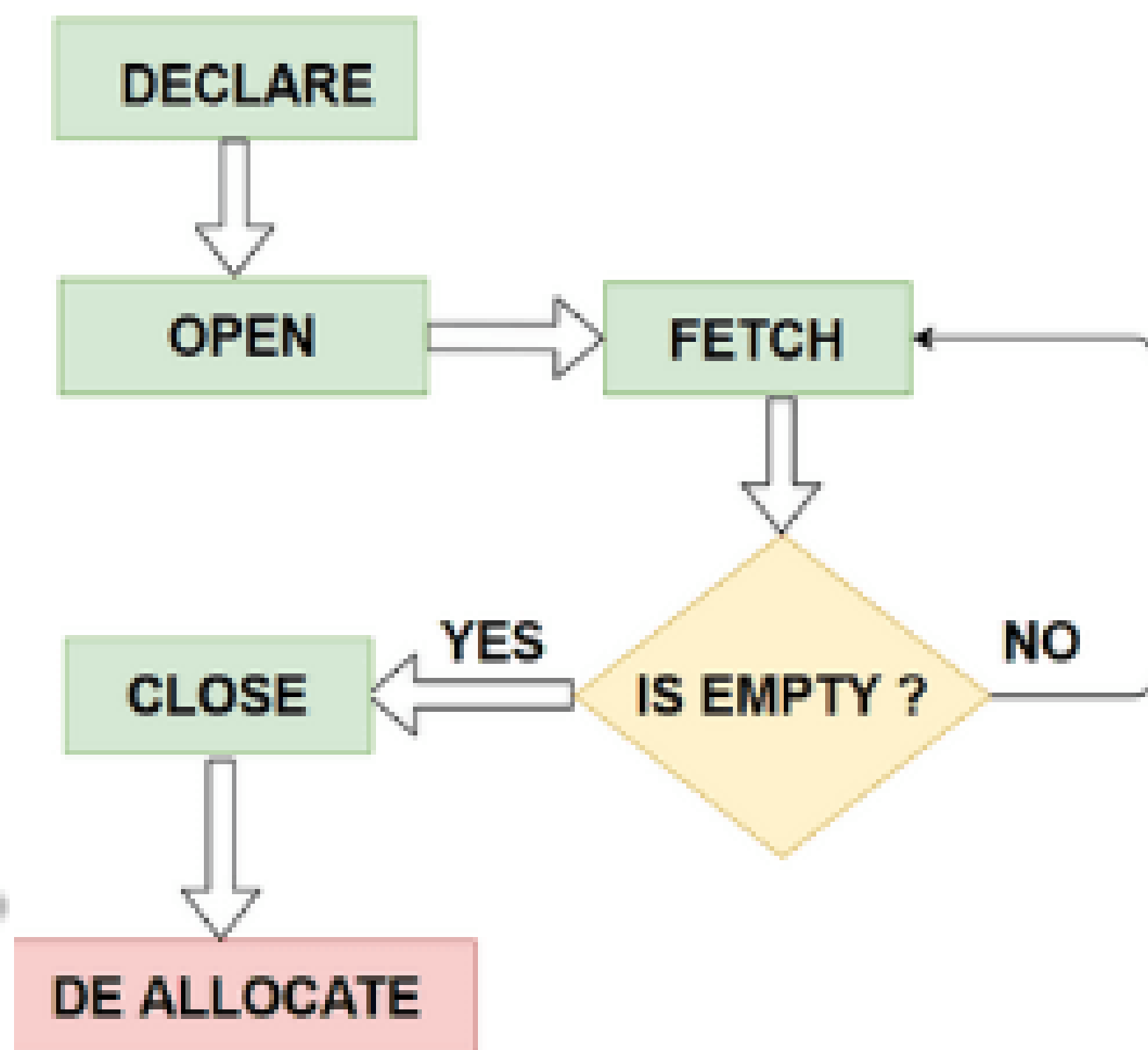
Hàm trả về biến bảng (Table variables)

Hàm trả về giá trị bảng (Table-valued)

SQL SERVER CURSOR



Microsoft®
SQL Server®



CURSOR trong SQL là gì?

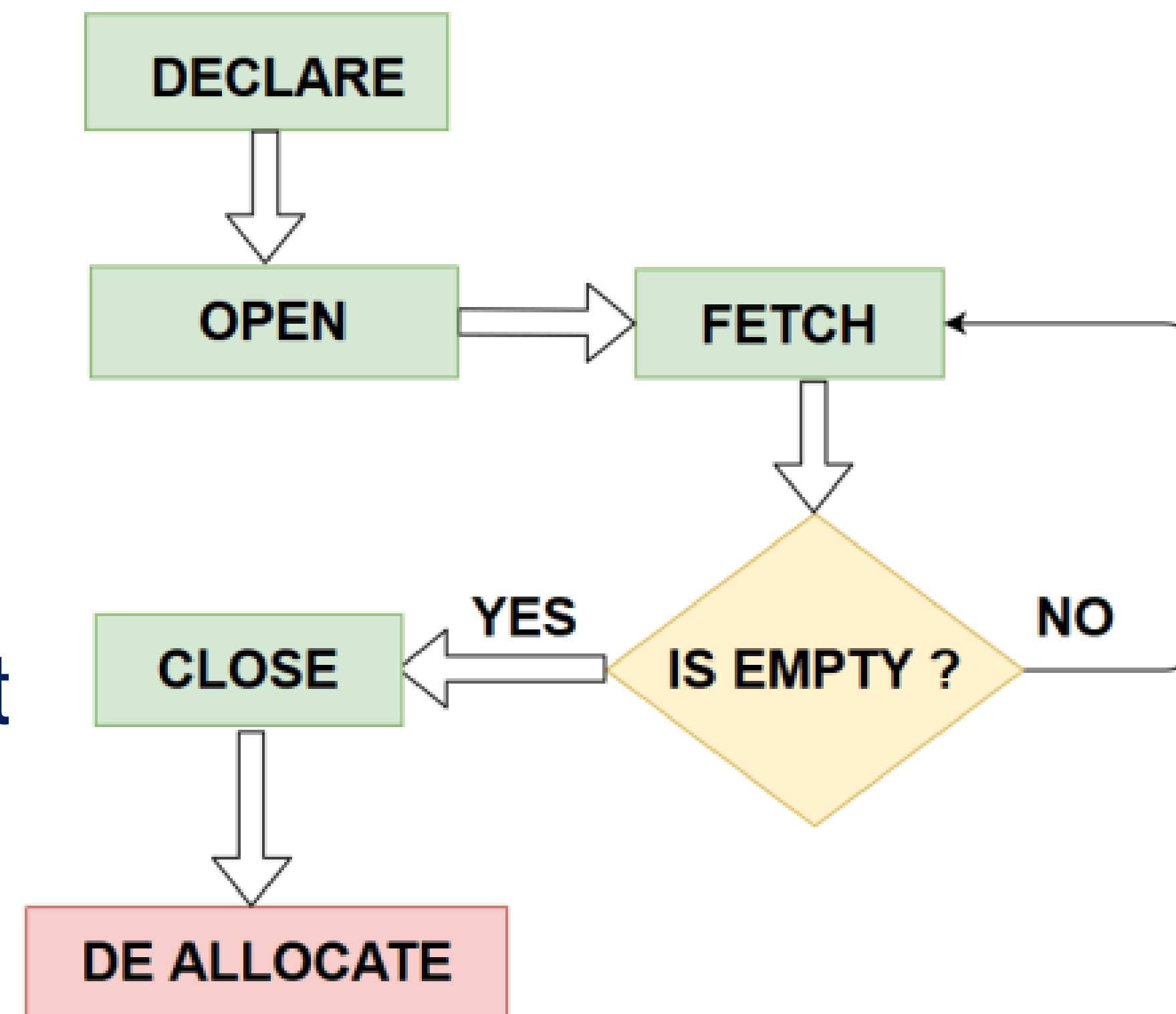
- In SQL Server, a **cursor** is a database object that allows us to retrieve and manipulate each row one at a time. It is nothing more than a row pointer.
- A cursor is always used alongside a **SELECT** command in SQL. It is typically a set of SQL logic that loops through a set number of rows one by one.

Tại sao phải sử dụng Con trỏ - CURSOR

- Duyệt các record ở table hay ở các kết quả từ câu lệnh truy vấn.
- Gồm 3 tính chất:
 - ✓ Read only
 - ✓ Non scrollable
 - ✓ Asensitive/Insensitive

Vòng đời của con trỏ được chia thành 5 giai đoạn:

- Declare Cursor (Khai báo con trỏ)
- Open Cursor (Mở con trỏ)
- Fetch Cursor (Lấy con trỏ)
- Close Cursor (Đóng con trỏ)
- Deallocate Cursor (Giải phóng con trỏ)



- Khai báo và liên kết một biến con trỏ với một câu lệnh SELECT nhất định.
- Cú pháp:

```
DECLARE cursor_name CURSOR FOR  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

- Mở con trỏ để có thể truy xuất các hàng.
- Cú pháp:

OPEN cursor_name;

- Truy xuất hàng tiếp theo từ tập kết quả của con trỏ.
- Cú pháp:
FETCH cursor_name **INTO** variable1, variable2, ...;

- ❑ Con trỏ đóng lại, giải phóng các tài nguyên được kết nối.
- ❑ Cú pháp:

CLOSE cursor_name;

- ❑ Định nghĩa con trỏ sẽ bị xóa và tài nguyên sẽ được giải phóng.
- ❑ Cú pháp:

DEALLOCATE cursor_name;

Hàm do người dùng định nghĩa

- Hàm để lấy các giá trị trả về thông qua tên hàm
- **Function chia 3 dạng:**
 - Hàm trả về giá trị **vô hướng** (scalar value): giá trị trả về có kiểu là kiểu của SQL như varchar, int,
 - Hàm trả về giá trị là **bảng tạm** (inline table-valued): table
 - Hàm trả về giá trị là **một bảng** nhưng trong hàm chứa nhiều lệnh

- Một hàm phải có tên. Tên không thể bắt đầu bằng một ký tự đặc biệt như @, \$, # hoặc ký tự tương tự khác.
- Lệnh SELECT là lệnh duy nhất hoạt động với các hàm.
- Dùng một hàm tại vị trí bất kỳ như AVG, COUNT, SUM, MIN, DATE và các hàm khác bằng truy vấn SELECT.
- Bất cứ khi nào gọi một hàm, nó sẽ được biên dịch.
- Hàm phải trả về một giá trị hoặc kết quả.
- Hàm chỉ sử dụng các tham số đầu vào.
- Bạn không thể dùng lệnh TRY và CATCH trong hàm.

- **Cú pháp**

CREATE FUNCTION tên_hàm ([danh_sách_tham_số])

RETURNS (kiểu_trả_về_của_hàm)

AS

BEGIN

các_câu_lệnh_của_hàm

END

Tạo hàm chuyển chuỗi ngày tháng năm sang chuỗi ký tự theo mã 112

Create Function Namthang(@d datetime)

Returns char(6)

As

Begin

Declare @st char(6)

Set @st=convert(char(6),@d,112)

Return @st

End

- **Cú pháp**

DROP FUNCTION function_name;

HÀM TRẢ VỀ GIÁ TRỊ VÔ HƯỚNG

❑ **Khái niệm:** Là hàm trả về một giá trị đơn lẻ (scalar value) như số, chuỗi, hoặc ngày tháng.

❑ **Cú pháp:**

```
CREATE FUNCTION function_name (parameters)
```

```
RETURNS data_type
```

```
AS
```

```
BEGIN
```

```
-- Câu lệnh thực thi
```

```
RETURN value
```

```
END
```

```
CREATE FUNCTION GetFullName (@firstName  
NVARCHAR(50), @lastName NVARCHAR(50))  
RETURNS NVARCHAR(100)
```

```
AS
```

```
    BEGIN    RETURN @firstName + ' ' + @lastName  
END
```

Sử dụng hàm:

```
SELECT dbo.GetFullName('John', 'Doe') AS FullName
```

❑ **Khái niệm:** Hàm trả về một biến bảng (table variable), cho phép lưu trữ nhiều dòng và nhiều cột dữ liệu.

❑ **Cú pháp**

```
CREATE FUNCTION function_name (parameters)
RETURNS @table_variable TABLE (column_definitions)
AS
BEGIN
    -- Câu lệnh thực thi
    INSERT INTO @table_variable
    SELECT ...
    RETURN
END
```

```
CREATE FUNCTION GetEmployeesByDepartment (@departmentId INT)
RETURNS @EmployeeTable TABLE (EmployeeID INT, EmployeeName
NVARCHAR(100))
AS
BEGIN
    INSERT INTO @EmployeeTable
    SELECT EmployeeID, EmployeeName
    FROM Employees
    WHERE DepartmentID = @departmentId
    RETURN
END
```

Sử dụng hàm:

```
SELECT * FROM dbo.GetEmployeesByDepartment(1)
```

❑ **Khái niệm:** Hàm trả về một giá trị bảng (table-valued), cho phép sử dụng như một bảng trong các truy vấn SQL.

❑ **Cú pháp**

```
CREATE FUNCTION function_name (parameters)
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN
```

```
(
```

```
-- Câu lệnh truy vấn
```

```
)
```

CREATE FUNCTION GetSalesByYear (@year INT)

RETURNS TABLE

AS

RETURN

(

SELECT ProductID, SUM(Amount) AS TotalSales

FROM Sales

WHERE YEAR(SaleDate) = @year

GROUP BY ProductID

)

- Sử dụng hàm:

SELECT * FROM dbo.GetSalesByYear(2023)

□ Cú pháp:

Create Function Ten_Ham[(các tham số)]

Returns @Ten_Bang_trả_về Table (Tên_cột Kiểu_dữ_liệu[,...])

AS

Begin

<các lệnh T-SQL>

Return

End

□ Chú ý:

- Cấu trúc bảng trả về bởi hàm được xác định dựa vào định nghĩa của bảng trong mệnh đề **Returns**. Biến @biến_bảng trong mệnh đề Returns có phạm vi sử dụng trong hàm và được sử dụng như một tên bảng.
- Câu lệnh Return không chỉ định giá trị trả về, giá trị trả về của hàm chính là các dòng dữ liệu có trong bảng tên @biến-bảng.

Nhanvien(MaNV,TenNV, Diachi)

Donhang(SOHD, MaNV, maKH, ngaydat, ngaygiao, Noigiao)

- Viết hàm đưa ra danh sách các nhân viên có mã nhân viên là tham số truyền vào @manv.
 - Nếu @manv = 0 thì đưa ra danh sách tất cả các nhân viên bán được hàng,
 - Ngược lại đưa ra danh sách nhân viên bán được hàng có mã bằng mã truyền vào

THỰC HÀNH



- ✓ Câu hỏi trắc nghiệm
- ✓ Bài tập

