



# Cursor

# Cursor là gì?

- ✧ Là 1 đối tượng của CSDL để giúp truy xuất và thao tác trên 1 bộ kết quả (result set) của lệnh truy vấn
- ✧ Cursor cho phép xử lý các hàng trong bộ kết quả theo 1 trong các cách sau:
  - Cho phép các hàng xác định được khôi phục từ bộ kết quả
  - Cho phép hàng hiện hành trong bộ kết quả được sửa đổi
  - Giúp di chuyển từ hàng hiện hành trong bộ kết quả đến 1 hàng khác
  - Cho phép dữ liệu được sửa đổi bởi các người dùng khác có thể thấy được sửa đổi này ngay trong bộ kết quả

# Cấu trúc của Cursor

∞ Các nhiệm vụ sau cần được thực hiện khi sử dụng cursor:

- Cursor cần được định nghĩa và các thuộc tính của nó cần được xác lập.
- Cursor cần được mở (open).
- Các hàng được yêu cầu cần được đọc (fetch) từ cursor.
- Dữ liệu trong hàng hiện hành của cursor có thể sửa đổi nếu cần.
- Cursor cần được đóng lại khi không dùng nữa.
- Cursor nên được giải phóng khỏi bộ nhớ sau khi đóng lại.

# Khai báo cursor

◦ *Declare Tên\_cursor CURSOR*

*[phạm vi] [di chuyển][trạng thái][xử lý]*

*For câu lệnh Select*

*[For update [OF danh sách cột]]*

**DECLARE *cursor\_name* CURSOR**

**[ LOCAL | GLOBAL ] -- Phạm vi**

**[ FORWARD\_ONLY | SCROLL ] --Di chuyển**

**[ STATIC | KEYSET | DYNAMIC | FAST\_FORWARD ]**

**-- Trạng thái**

**[ READ\_ONLY ] – Xử lý**

**FOR *select\_statement***

**[ FOR UPDATE [ OF *column\_name* [ ,...*n* ] ] ]**

# Khai báo cursor

## Phạm vi

- ∞ **LOCAL**: phạm vi của cursor là cục bộ trong phạm vi 1 batch, thủ tục, hay trigger trong đó cursor được tạo ra.
- ∞ **GLOBAL**: phạm vi của cursor là toàn cục được dùng trong suốt phiên kết nối. Cursor sẽ bị xóa khi mất kết nối

# Khai báo cursor

## Di chuyển

∞ **Forward-only**: chỉ có thể đọc (fetch) các hàng một cách tuần tự từ đầu đến cuối bộ kết quả.

∞ **SCROLL**: cho phép sử dụng tất cả các tùy chọn của lệnh đọc (FIRST, LAST, PRIOR, NEXT, RELATIVE, ABSOLUTE)

# Lệnh mở cursor

☞ Để mở cursor

**OPEN cursor\_name**

# Đọc dữ liệu - Fetching data

∞ **cú pháp:**

**FETCH** [[**NEXT** | **PRIOR** | **FIRST** | **LAST** |  
**ABSOLUTE** *n* | **RELATIVE** *n*] ]

**FROM** *cursor\_name* [ **INTO** *@variable\_name* [ ,...*n* ] ]

- ∞ **NEXT**: chuyển cursor tới hàng kế tiếp trong kết quả (bộ mặc định)
- ∞ **PRIOR**: quay về lại hàng kế trước trong bộ kết quả.
- ∞ **FIRST**: quay về lại hàng đầu tiên.
- ∞ **LAST**: chuyển đến hàng cuối cùng trong bộ kết quả.
- ∞ **ABSOLUTE** *n*: chuyển đến hàng thứ *n*.
- ∞ **RELATIVE** *n*: chuyển đến hàng thứ *n* kể từ hàng hiện hành.



## Biến @@fetch\_status

☞ Nó trả về trạng thái của lệnh fetch vừa được thực thi.

<b><i>@@fetch_status</i></b>	<b><i>Description</i></b>
0	Successful execution of the fetch statement.
-1	Unsuccessful execution of the fetch statement.
-2	Row being fetched is missing.

# Đóng và giải phóng Cursor khỏi bộ nhớ

- ° Phải đóng cursor lại trước khi giải phóng nó khỏi bộ nhớ.
  - ✧ Cú pháp lệnh đóng cursor
- CLOSE cursor\_name**
- ✧ Khi lệnh CLOSE được thực thi, cấu trúc của cursor vẫn còn và có thể mở lại được.
  - ✧ Cú pháp lệnh giải phóng cursor khỏi bộ nhớ

**DEALLOCATE cursor\_name**



# Thứ tự các thao tác khi xử lý dữ liệu trên Cursor

1. Định nghĩa biến Cursor
2. Mở Cursor
3. Duyệt và xử lý dữ liệu trên Cursor
4. Đóng và giải phóng Cursor

# Ví dụ: Viết SP in danh sách Khoa dùng lệnh print

```
CREATE PROCEDURE SPTestCursor
AS
BEGIN
    DECLARE @makhoa varchar(10)
    DECLARE @tenkhoa nvarchar(100)
    --Khai báo cursor
    DECLARE cur_khoa cursor
    FOR
        SELECT ma,tenKhoa
        FROM Khoa
    --Mở cursor
    OPEN cur_khoa
    --Đọc dữ liệu dòng đầu tiên vào @makhoa và @tenkhoa
    FETCH cur_khoa into @makhoa, @tenkhoa
    -- Thực hiện vòng lặp để hiển thị tất cả các hàng của cursor.
    While (@@fetch_status = 0)
    BEGIN
        Print 'Ma Khoa = ' + @makhoa
        Print 'Ten Khoa = ' + @tenkhoa
        -- Đọc vào hàng kế tiếp của cursor.
        FETCH cur_khoa into @makhoa, @tenkhoa
    END
    -- Đóng cursor
    CLOSE cur_khoa
    -- Giải phóng cursor khỏi bộ nhớ.
    DEALLOCATE cur_khoa
END
```

# Bài tập

## Dùng CSDL Quản lý sinh viên

1. Viết stored procedure, dùng lệnh print in ra danh sách sinh viên
2. Viết stored procedure, dùng lệnh print in ra danh sách các môn học
3. Viết stored procedure in điểm các môn học của sinh viên có mã số là maSinhVien được nhập vào. (Chú ý: điểm của môn học là điểm thi của lần thi sau cùng). Các môn chưa có điểm thì ghi điểm là <chưa có điểm>



## Dùng CSDL Quản lý sinh viên

1. Viết function tính điểm thi sau cùng của một sinh viên trong một môn học cụ thể
2. Viết store procedure in ra điểm trung bình của một sinh viên (chú ý: điểm trung bình được tính dựa trên lần thi sau cùng), sử dụng function ở câu trên.