CÁC HỆ QUẢN TRỊ CSDL

ĐẠI HỌC SƯ PHẠM TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
Phiên bản 2019

CHƯƠNG 3:

LẬP TRÌNH VỚI CURSOR

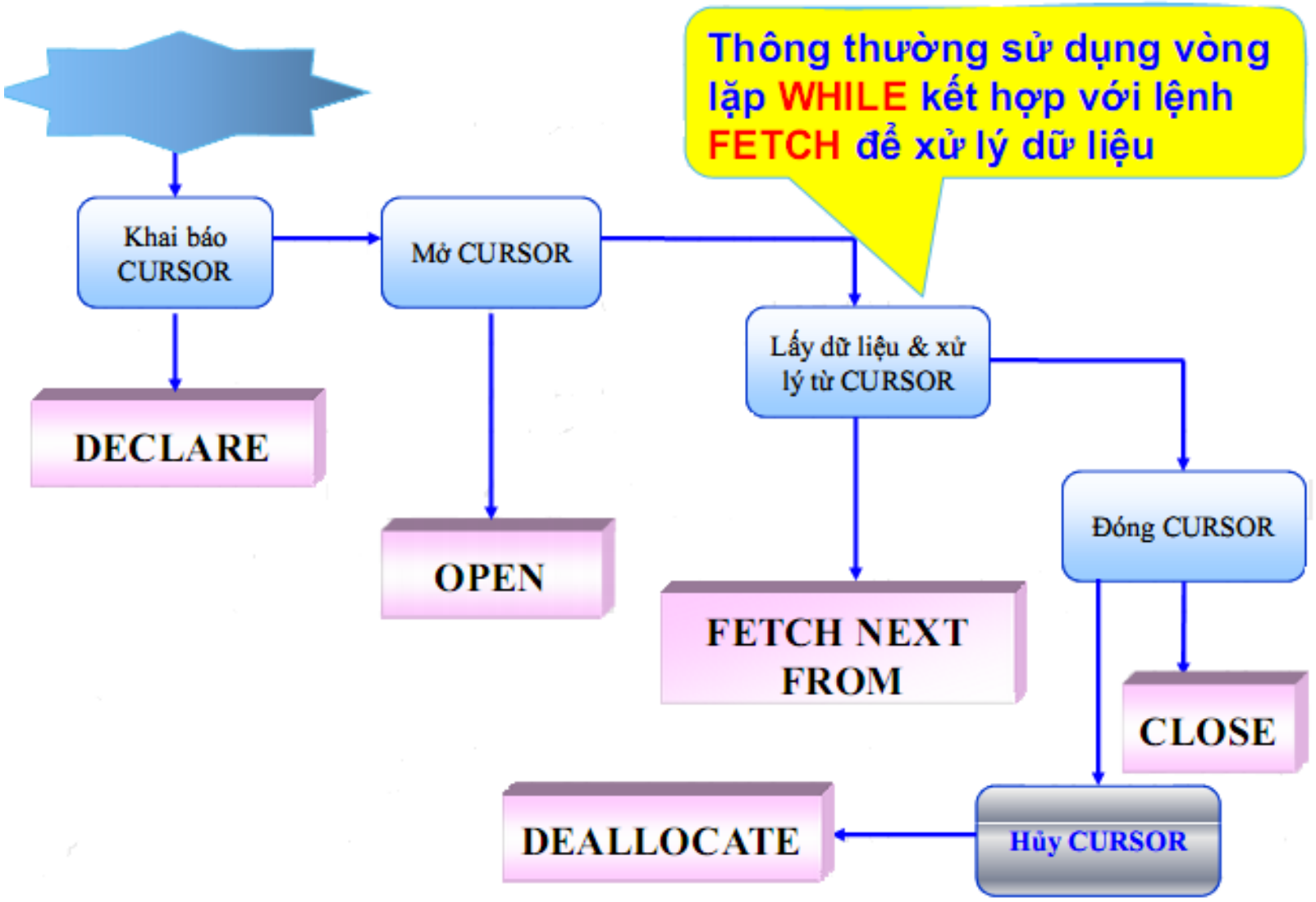
Lập trình với con trỏ

- Con trỏ là một đối tượng cơ sở dữ liệu được sử dụng bởi ứng dụng để thao tác với các hàng dữ liệu thay vì các tập hợp dữ liệu.
- Con trỏ được dùng với Procedure và Trigger

Lập trình với con trỏ

- Với con trỏ chúng ta có thể:
  - Cho phép định vị các hàng chỉ định của tập kết quả.
  - Nhận về một hàng đơn hoặc tập hợp các hàng từ vị trí hiện tại của tập kết quả.
  - Hỗ trợ sửa đổi dữ liệu của hàng ở vị trí hiện tại trong tập kết quả.
  - Hỗ trợ nhiều cấp độ quan sát đối với các thay đổi được tạo ra bởi các người dùng khác trên các dữ liêu của tập kết quả.

Quy trình xử lý con trỏ



Tạo con trỏ

- Lệnh DECLARE dùng để tạo một con trỏ.
- Nó chứa các lệnh SELECT để bao gồm các bản ghi từ bảng.
- Cú pháp:

```
DECLARE  <Cursor_Name> CURSOR
FOR          <Select Statements>
[FOR UPDATE [OF Column_name[,….N]]]
```

Tạo con trỏ

- Cú pháp đầy đủ:

```
DECLARE  <Cursor_Name> CURSOR
[LOCAL | GLOBAL]
[FORWARD ONLY | SCROLL]
[STATIC | KEYSET | DYNAMIC |FAST_FORWARD]
[READ_ONLY | SCROLL_LOCKS | OPTIMISTIC]
[TYPE_WARNING]
FOR <Select Statements>
[FOR UPDATE [OF Column_name[,….N]]]
```

Trong đó

- Phạm vi: [LOCAL | GLOBAL]
  - Local :chỉ sử dụng trong phạm vi khai báo(mặc định)
  - Global :sử dụng chung cho cả kết nối
- Di chuyển: [FORWARD ONLY | SCROLL]
  - ForWard_Only :chỉ di chuyển một hướng từ trước ra sau(mặc định)
  - Scroll : di chuyển tùy ý

Trong đó

- Trạng thái: [STATIC | KEYSET | DYNAMIC]
  - Static : dữ liệu trên Cursor không thay đổi mặt dù dữ liệu trong bảng nguồn thay đổi (mặc định)
  - Dynamic :dữ liệu trên Cursor sẽ thay đổi khi dữ liệu trong bảng nguồn thay đổi
  - KeySet :giống Dynamic nhưng chỉ thay đổi những dòng bị cập nhật

Trong đó

- Xử lý [READ_ONLY | SCROLL_LOCKS]
  - Read_Only :chỉ đọc(mặc định)
  - Scroll_Lock : đọc/ghi
- Câu lệnh select :không chứa các mệnh đề Into,Compute,Compute by
- Danh sách cột cập nhật : là danh sách các cột sẽ thay đổi được

Các bước sử dụng con trỏ

- Mở con trỏ:

  OPEN <Cursor_name>

- Duyệt và xử lý dữ liệu trong cursor :

  FETCH <Cursor_name>

- Đóng con trỏ:

  CLOSE <Cursor_name>

- Xoá các tham chiếu tới con trỏ:

  DEALLOCATE <Cursor_name>

Truy xuất và duyệt con trỏ

- Cú pháp :

FETCH Hướng di chuyển From Tên_biến_Cursor Into DSbiến

Ví dụ:

```
declare Cur_MatHang CurSor
for     select MaMH,tenmh from MatHang
open Cur_MatHang
declare @maMH char(4),@tenMH varchar(100)
while 0=0
      begin
            fetch next from Cur_MatHang into @maMH,@tenMH
            if @@fetch_status<>0 break
            print 'Mã mặt hàng :' + @maMH +' Tên mặt hàng :' +
            @tenMH
      end
close Cur_MatHang
deallocate Cur_MatHang
```

Truy xuất và duyệt con trỏ

- FETCH FIRST: Truy xuất hàng đầu tiên.
- FETCH NEXT: Truy xuất hàng tiếp theo hàng truy xuất trước đó.
- FETCH PRIOR: Truy xuất hàng trước hàng truy xuất trước đó.
- FETCH LAST: Truy xuất hàng cuối cùng.

Truy xuất và duyệt con trỏ

- FETCH ABSOLUTE *n*:

  Di chuyển đến mẩu tin thứ n tính từ mẩu tin đầu tiên

  - Nếu n là một số nguyên dương, nó sẽ truy xuất n hàng trong con trỏ.
  - Nếu n là một số nguyên âm, n hàng trước hàng cuối cùng trong con trỏ được truy xuất.
  - Nếu n bằng 0, không hàng nào được truy xuất.
  - Ví dụ, FETCH Absolute 2 sẽ hiển thị bản ghi thứ hai của một bảng.

- FETCH RELATIVE *n*:

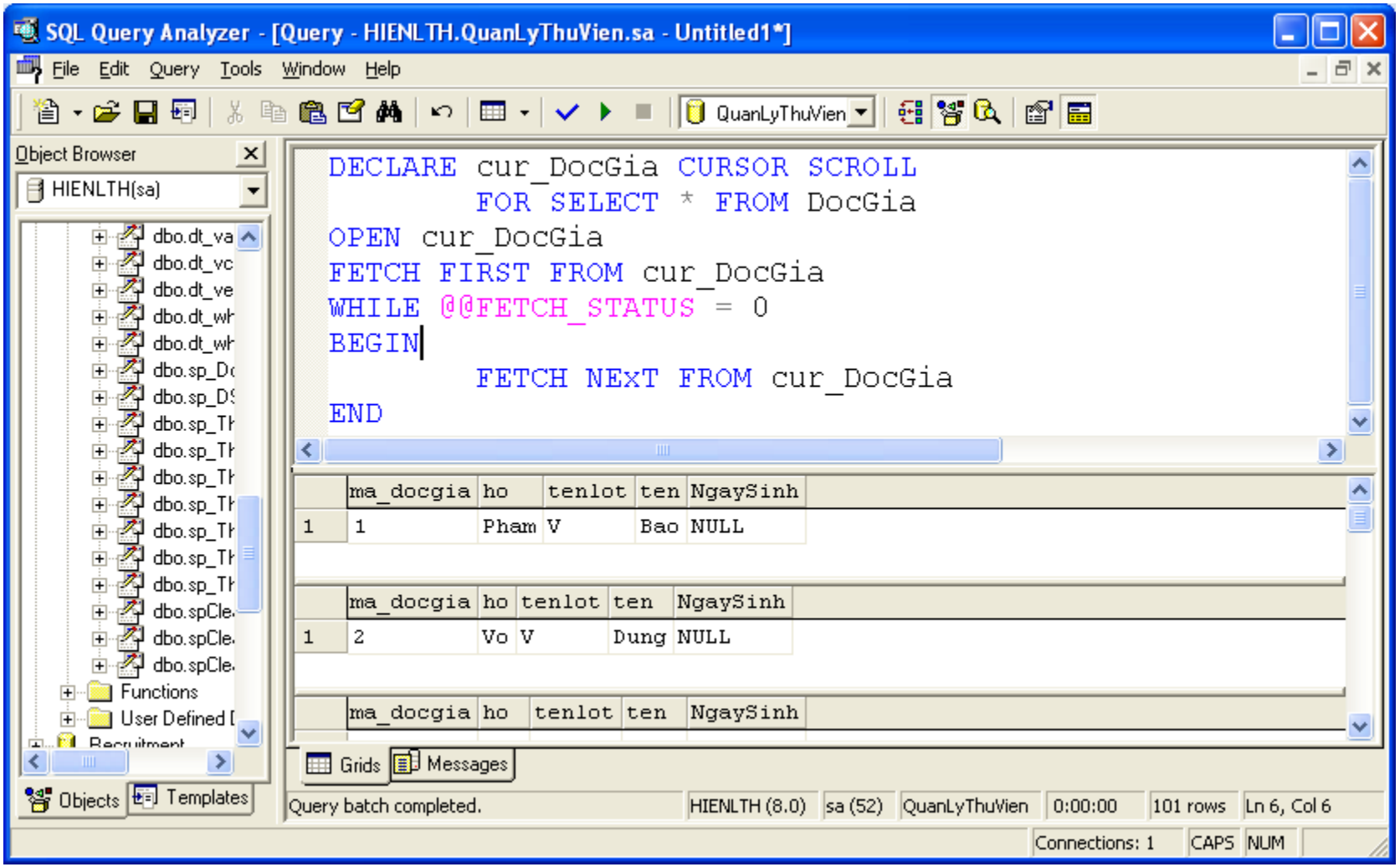
  Di chuyển đến mẩu tin thứ n tính từ mẩu tin hiện hành

  - Nếu *n* là số âm, *n* hàng trước hàng truy xuất trước đó được truy xuất.
  - Nếu *n* bằng 0, hàng hiện tại được nhận về.

Các biến toàn cục của lệnh FETCH

- @@FETCH _STATUS: Biến này trả về một số nguyên biểu diễn kết quả của lệnh truy xuất cuối cùng của con trỏ.
  - **@@FETCH_STATUS nếu <>0 thất bại**
  - **@@FETCH_STATUS nếu = 0 thành công**
- @@CURSOR_ROWS: Biến này trả về tổng số hàng hiện tại trong con trỏ đang mở.

Ví dụ tạo con trỏ



Con trỏ (tt)

- Một con trỏ là một đối tượng cơ sở dữ liệu được sử dụng bởi ứng dụng để thao tác với các hàng dữ liệu thay vì các tập hợp dữ liệu. Sử dụng con trỏ, nhiều tác vụ có thể được thực hiện theo từng hàng trên tập kết quả mà có thể cần hoặc không cần sự có mặt của bảng gốc

Con trỏ (tt)

- Con trỏ được tạo bằng lệnh **DECLARE**. Đầu tiên con trỏ được khai báo và tạo ra trong bộ nhớ. Sau đó nó mới được mở.
- Lệnh **OPEN** mở con trỏ. Việc nhận về các bản ghi từ một con trỏ được gọi là fetching. Một người dùng chỉ có thể nhận về một bản ghi tại một thời điểm.
- Lệnh **FETCH** được sử dụng để đọc các bản ghi từ con trỏ.

Con trỏ (tt)

- Ngầm định, 1 con trỏ là **forward only**. Nó có thể truy xuất tuần tự các bản ghi từ bản ghi đầu tiên đến bản ghi cuối cùng. Nó không thể truy xuất trực tiếp hàng thứ 1 hoặc hàng cuối cùng trong một bảng.
- Khi 1 con trỏ tạm thời không cần thiết, nó có thể được đóng bởi lệnh **CLOSE**.
- Mỗi khi con trỏ không được sử dụng, các tham chiếu đến nó nên được loại bỏ bằng lệnh **DEALLOCATE**

STORED PROCEDURE

Stored Procedure

- Cho phép lập trình theo hướng Module
- Thực thi nhanh hơn, giảm được việc chiếm dụng đường truyền mạng
- Bảo mật
- Xử lý các chức năng và chia sẽ với các ứng dụng khác

Stored Procedure

- ***Cú pháp:***

```
CREATE PROCEDURE proc_name
AS
BEGIN
                    sql_statement1
                    sql_statement2
END
```

Stored Procedure Syntax

```
CREATE PROCEDURE StoredName
@Parameter1 DataType [=DefaultValue,]
@Parameter2  DataType OUTPUT,
@Parameter3  DataType OUTPUT
AS
BEGIN
      BEGIN TRANSACTION
            {T-SQL Statement1}
            If  @Error <> 0
                  Goto Err_Handle
            {T-SQL Statement2}
            If  @Error <> 0
                  Goto Err_Handle
      COMMIT TRANSACTION
      Return(0)
      Err_Handle:
            ROLLBACK TRANSACTION
                  Return(@Error)
END
```

Ví dụ 1 – SP không tham số

```
CREATE PROCEDURE  sp_XemDSSV
AS
BEGIN
    PRINT N'DANH SÁCH SINH VIÊN'
    SELECT MSSV, HoLot, Ten, NgaySinh,
        NoiSinh, DiaChi
    FROM SinhVien
END
```

Ví dụ 2 – SP có tham số

```
CREATE PROCEDURE  sp_XemSV
        @MaSV nvarchar(11)
AS
BEGIN
    PRINT N'SINH VIÊN'
    SELECT HoLot, Ten, NgaySinh,
        NoiSinh, DiaChi
    FROM SinhVien
    WHERE MSSV = @MaSV
END
```

Xem nội dung SP

- *Cú pháp:*

  sp_helptext proc_name

Ví dụ:

- *Mở Query Analyzer, gõ:*

  *sp_helptext **sp_XemDSSV***

  *sp_helptext **sp_XemSV***

- *Kiểm tra chính tả, nội dung procedure.*

Gọi Stored Procedure

- ***Cú pháp:***

  EXECUTE proc_name danh_sách_tham_số

  hoặc

  EXEC proc_name danh_sách_tham_số

  hoặc

  proc_name danh_sách_tham_số

//Mỗi tham số các nhau một dấu phẩy

Ví dụ

- ***Mở Query Analyzer, gõ:***

  EXECUTE sp_XemDSSV

  EXECUTE sp_XemSV 'K29.103.010'

  hoặc

  EXEC sp_XemDSSV

  EXEC sp_XemSV 'K29.103.010'

  hoặc

  sp_XemDSSV

  sp_XemSV 'K29.103.010'

- ***Bấm F5 để thực thi***