Stored Procedure

Mục tiêu bài học

- Hiểu được stored procedure là gì, procedure hoạt động như thế nào.
- Quản lý procedure: Tạo, xoá, sửa và thực thi
- Tham số trong store procedure
- Bài tập áp dụng

Nội dung bài học

1. Giới thiệu Stored procedures
2. Tạo, thực thi, cập nhật và xoá stored procedures
3. Bài tập thực hành
4. Truyền tham số trong stored procedures
5. Điều khiển lỗi
6. Một số lưu ý

Giới thiệu Stored Procedure (1)

- Stored procedure là một tập các lệnh Transact SQL được đặt tên và lưu trữ trong database server
- Có thể nhận tham số vào và tham số trả giá trị về
- Trả về trạng thái thực thi của procedure là thành công hay không thành công

Giới thiệu Stored Procedure (2)

- Có 5 loại stored procedure:
  - System (sp_): có trong master database, được truy xuất từ bất kỳ một database nào, nhằm cung cấp các thông tin system catalog hoặc thực hiện các nhiệm vụ của administration.
  - Local : được tạo từ user
  - Temporary: có tên bắt đầu bằng # (local) hoặc ## (global). Không còn tồn tại sau khi SQL Server shutdown
  - Remote: giới hạn việc thực hiện một stored procedure trên remote SQL Server
  - Extended (xp_) được thực hiện bởi các ngôn ngữ khác và được gọi là các DLL. Sau khi viết xong extended stored procedure, **sysadmin** đăng ký extended stored procedure với SQL Server và sau đó gán quyền cho users khác để thực hiện. Extended stored procedures chỉ được có trong **master** database.

Xử lý Stored procedure (1)_Initial

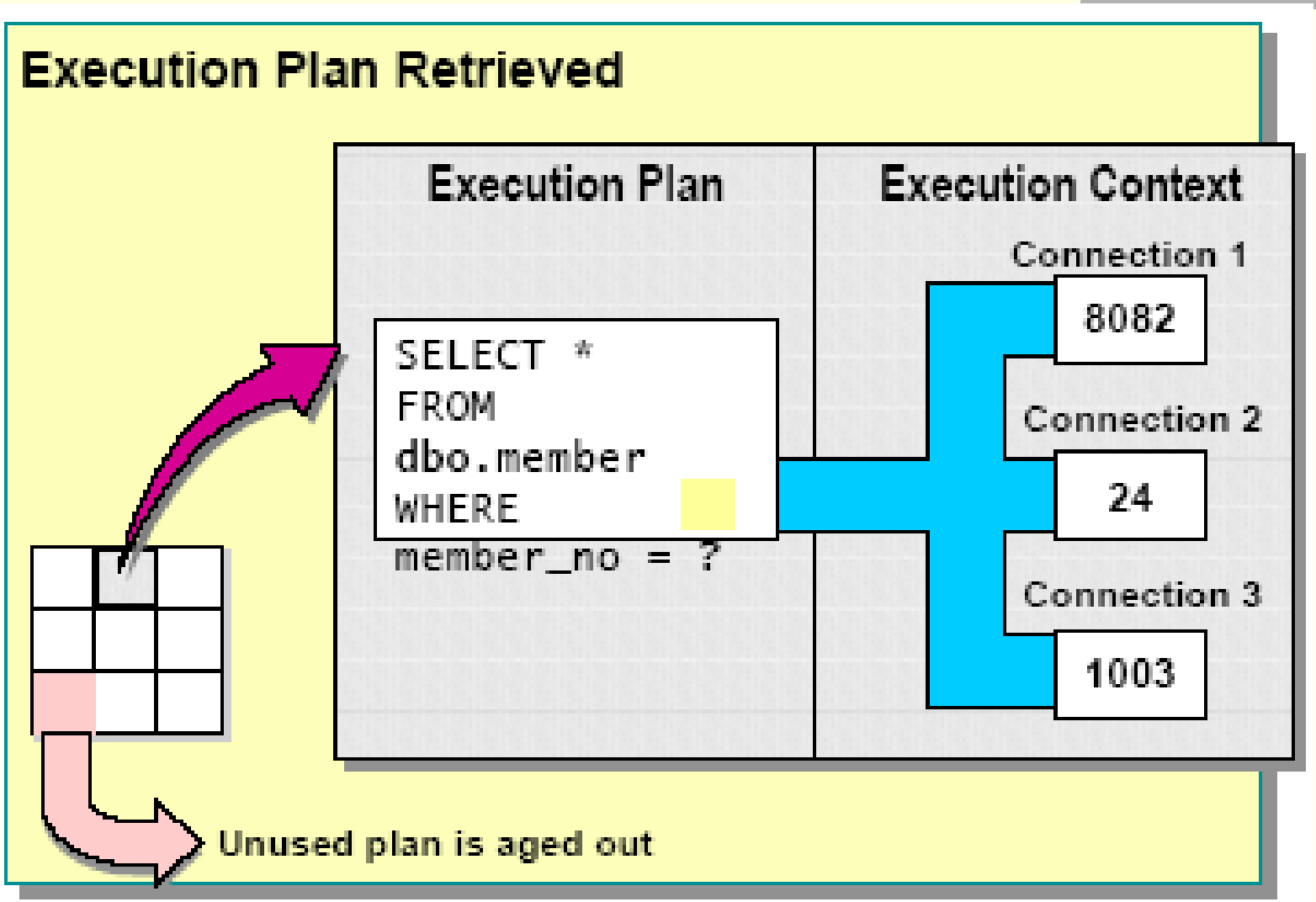
Execution: Khi lần thứ nhất mà procedure được thực hiện hoặc khi procedure phải recompile, query processor sẽ đọc procedure trong process được gọi là resolution.

Sau giai đoạn resolution, SQL Server sẽ tạo ra một sơ đồ thực hiện và đặt sơ đồ này trong procedure cache

Creation: Các lệnh trong procedure sẽ được phân tích cú pháp theo cú pháp của T-SQL. Nếu thành công, tên của stored procedure được lưu trong SysObjects table, còn text của procedure lưu trong SysComments.

Delayed Name Resolution: cho phép stored procedure tham chiếu đến các đối tượng chưa tồn tại trong lúc procedure được tạo. Delayed Name resolution được thực hiện trong lúc procedure được thực hiện

Xử lý stored procedure (2)_ Subsequent



Lợi ích của stored procedure

- Cho phép lập trình theo hướng modular (modular programming)
- 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

Cú pháp tạo procedure

CREATE PROCEDURE *procedure_name*

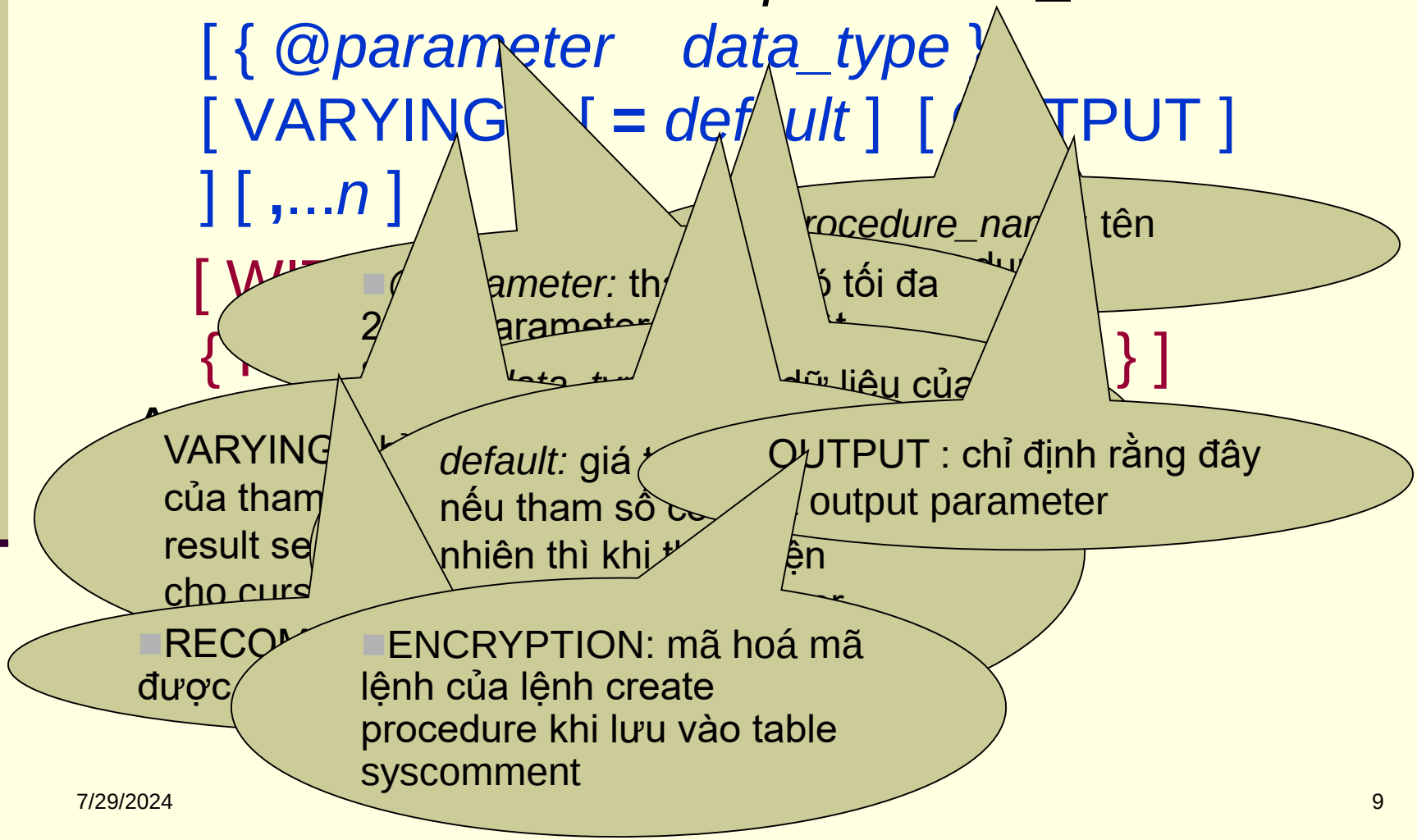
[{ *@parameter* *data_type* }

[VARYING] [= *default*] [OUTPUT]

] [,...*n*]

[WI

{ [



Ví dụ: CSDL Northwind

- Tạo procedure P1 để liệt kê danh sách tất cả các products

```
Create procedure P1
 as
    Select * from Product
Exec  P1
```

- sản phẩm tăng 10%

Create procedure

Lưu ý

- Có thể tham chiếu đến các tables, view, procedure khác cũng như các temporary table
- Để tạo một procedure, user phải có quyền CREATE PROCEDURE (sysadmin, hoặc database owner)
- Kích thước của procedure tối đa là 128 MB
- Có thể lồng 32 cấp procedure
- Dùng procedure sp_helptext để hiển thị nội dung text của stored procedure mà user đã tạo
- Không thể kết hợp lệnh create procedure với các lệnh SQL khác để tạo thành một bó lệnh (batch)
- Chỉ có thể tạo procedure trong database hiện hành.

Thực thi procedure

Lệnh để thực thi một stored procedure:

EXECUTE

[*@return_status* =] *procedure_name*

[[*@parameter* =] { *value* | *@variable*

[OUTPUT] | [DEFAULT]]

[,...n]

[WITH RECOMPILE]

Chỉ định rằng lấy giá trị trả về của tham số

Chỉ định rằng lấy giá trị default của tham số

Chỉ định rằng procedure phải recompile trước khi thực hiện

Ví dụ

- Thực thi procedure P1 và P2:

Execute P1

go

Execute P2

go

Ví dụ 2:

```
Create procedure Mexico_Customers
 as
    select * from customers
      where country='Mexico'
go
```

```
Execute Mexico_customers
```

Procedure có tham số

```
create proc CustomerListOfCountry
                        @country varchar(40)
as
select customerid, CompanyName from customers
   where country=@country
 go
```

execute CustomerListOfCountry **'Canada'**

Hoặc truyền tham số với giá trị khác

execute CustomerListOfCountry 'USA'

Nếu không truyền tham số:

Execute CustomerListOfCountry ????????

Procedure có tham số có giá trị default

```
create proc CustomerList @country
   varchar(40)='canada'
   as
   select customerid, CompanyName from
   customers where country=@country
 go
```

Gọi thực thi có : **execute** CustomerList **‘Mexico’**

Gọi thực thi không truyền tham số:

Execute CustomerList

Procedure dùng output parameter

Tạo Procedure để trả về số lượng khách hàng có giá trị country là tham số truyền vào:

```
create proc P2
        @country varchar(40) = '%',  @total integer OUTPUT
AS
    SELECT @total = count(*) FROM customers WHERE
    country  like  @country
Go
```

Thực thi procedure P2

```
declare @sluong  integer
Execute  P2 'canada', @sluong output
SELECT 'The total customers of  canada is '
   +str(@sluong,4)
go
```

Tham số có kiểu là cursor

- Chỉ dùng cho tham số OUTPUT.
- Nếu kiểu của tham số là **cursor** thì VARYING và OUTPUT là các từ khóa bắt buộc.
- Nếu VARYING được chỉ định cho một tham số thì kiểu dữ liệu của tham số phải là **cursor** và từ khoá OUTPUT phải được chỉ định.

Kiểu dữ liệu Cursor (1)

- Được dùng trong procedure hoặc trigger
- Chứa result set column, record
- Xử lý cursor:
  - Khai báo biến kiểu cursor chứa dữ liệu trả về
  - Kết hợp cursor với câu lệnh select bằng lệnh DECLARE CURSOR
  - Dùng lệnh OPEN để mở cursor
  - Dùng lệnh FETCH INTO để đổ một record hiện hành vào các biến tương ứng với từng column.
  - Dùng lệnh CLOSE để đóng cursor

Kiểu dữ liệu Cursor (2)

- sp_cursor_list để lấy ra danh sách các cursor hiện có
- sp_describe_cursor, sp_describe_cursor_column và sp_describe_cursor_tables để xem đặc tính của cursor
- Sau khi cursor mở, hàm @@CURSOR_ROWS tra về số lượng record
- Sau lệnh FETCH, hàm @@FETCH_STATUS để phản ánh trạng thái fetch sau cùng (0,-1)

System stored procedure	Description
sp_cursor_list	Returns a list of cursors currently visible on the connection and their attributes.
sp_describe_cursor	Describes the attributes of a cursor, such as whether it is a forward-only or scrolling cursor.
sp_describe_cursor_columns	Describes the attributes of the columns in the cursor result set.
sp_describe_cursor_tables	Describes the base tables accessed by the cursor.

Cú ph

Cursor

Global: Biến có phạm vi Global, có thể truy xuất trong bất kỳ procedure

DYNAMIC: Định nghĩa một cursor phản ánh tất cả các

READ_ONLY
cursor

OPTIMISTIC: SQL Server không lock các dòng nếu như chúng được đọc vào cursor.

FOR UPDATE [OF *column_name* [,...*n*]]: định nghĩa các cột có thể cập nhật trong cursor.

[STATIC]
Local: Biến **phạm**
local
FAST_FORWARD
[READ_ONLY]
OPTIMISTIC

FORWARD_ONLY: Cursor chỉ c chuyển một chiều từ dòng đầu đế dòng cuối

SCROLL_LOCKS: khoá các dữ liệu mà đã được đọc vào cursor

trong
các khoá trong

cursor được lưu trong một table trong database **tempdb** gọi là **keyset**.

Ví dụ 1

```
DECLARE customer_cursor CURSOR
        FOR SELECT * FROM customers
OPEN customer_cursor  -- mở cursor
FETCH NEXT FROM customer_cursor
```

Ví dụ 2 (1)

```
DECLARE @customerId varchar(11),
   @CompanyName varchar(30), @message
   varchar(80)
PRINT "-------- Customer report --------“
DECLARE customer_cursor CURSOR
   FOR SELECT customerid, companyName
   FROM customers WHERE country = "USA"
OPEN customer_cursor
FETCH NEXT FROM customer_cursor INTO
   @customerid, @companyName
```

Ví dụ 2 (2)

```
While @@FETCH_STATUS = 0
   begin
      print  'Customer ID:    ' + @customerID
      print 'Company Name: ' + @companyName
     Fecth next from customer_cursor into
          @customerid, @companyName
 end
Close customer_cursor
Deallocate customer_cursor
go
```

Sử dụng OUTPUT cursor parameter

```
USE northwind
CREATE PROCEDURE customer_cursor
   @customer_cursor CURSOR VARYING
   OUTPUT AS
SET @customer_cursor = CURSOR
   FORWARD_ONLY STATIC
FOR
SELECT * FROM  CUSTOMERS
OPEN @customer_cursor
GO
```

Sử dụng tham số cursor trả về

```
DECLARE @MyCursor CURSOR
EXEC customer_cursor  @customer_cursor =
   @MyCursor OUTPUT
WHILE (@@FETCH_STATUS = 0)
BEGIN
   FETCH NEXT FROM @MyCursor
END
CLOSE @MyCursor
DEALLOCATE @MyCursor
```

Bài tập ứng dụng (NorthWind)

- Tạo procedure và thực thi để in ra company name có số lượng orders nhiều nhất
- Tạo proc p1 để trả về doanh thu của năm truyền vào, nếu user không truyền ngày vào thì lấy năm hiện hành
- Khai báo một procedure CustomersOfCountryCursor để lấy ra một cursor chứa các record của table customers có country bằng giá trị truyền vào. Thực thi CustomersOfCountryCursor và in ra các dữ liệu có trong cursor trả về.

Bài tập áp dụng (database QLVT)

1. Tạo procedure P1 để lấy ra danh sách các hoá đơn gồm các thông tin: MAHD, NGAY, TENKH, TONGTG
2. Tạo procedure P2 để xoá các chi tiết hoá đơn của hoá đơn có mã là tham số truyền vào
3. Tạo procedure P3 để tính tổng doanh thu của năm với năm là tham số truyền vào và trả về giá trị là tổng doanh thu đã tính được.
4. Truyền vào tháng, năm và trả về lợi nhuận trong tháng và năm đó.
5. Lấy ra mặt hàng x bán có lãi ít nhất.

```
create proc p3 @nam int, @dt bigint output
as
  set @dt=(select sum(sl*giaban)
  from HOADON a, CHITIETHOADON b
  where a.MAHD=b.MAHD and YEAR(NGAY)=@nam)
go
```

- declare @doanhthu bigint
- exec p3 2000, @doanhthu output
- select 'doanh thu nam 2000 = ' + STR(@doanhthu, 10)`

Tóm tắt nội dung buổi học

- Stored procedure trong SQL Serever giống procedure trong các ngôn ngữ lập trình
- Xử lý nhanh hơn batch
- Procedure có thể có các tham số input và output
- thực thi một stored procedure dùng lệnh execute

- Tạo proc lấy ra danh sách khách hang có dịa chị là tham số truyền vào:

```
CREATE PROC P2 @DC VARCHAR(50)
AS
   select * from khachhang where diachi=@dc
```

```
Create proc P1
As
Select hd.mahd, tenkh, ngay, sum(sl*giaban) as tongtg
From hoadon hd, khachhang kh, chitiethoadon cthd
Where (hd.mahd=cthd.mahd)  and
            (kh.makh = hd.makh)
Group by hd.mahd, tenkh, ngay
```

```
Create proc p2 @mahd nvarchar(10)
As
   delete *  from  chitiethoadon where mahd=@mahd
Go
-- thuc thi
Exec p2 ‘hd002’
go
```

```
Create proc p3 @nam int, @dt int OUTPUT
As
   select  @dt=sum(sl*giaban)
      From chitiethoadon cthd, hoadon hd
      Where  (hd.mahd=cthd.mahd)  AND
               year(ngay)= @nam
Go
Declare @dt int
Exec p3 2000,@dt OUTPUT
Print ‘Doanh thu nam 2000 la ‘ + str(@dt,8)
```

Trong QLVT

- Tạo procedure p6 để lấy ra makh và tên của các khách hàng đã mua hàng trong tháng …. Và năm ….. (tham số input). Danh sách này được trả về trong một kiểu cursor.
- Thực thi P6 để lấy ra danh sách các khách hàng của tháng 6 năm 2000 và in ra tên của các khách hàng đó.

BT: tao proc P7 nhận vào makh xuất ra mavt, tenvt và số lượng vật tư đó mà kh đã mua. Sử dụng cursor để xuất kết quả ra màn hình

- alter proc P6 @t int, @n int, @cur cursor varying output
- as
- SET @cur = cursor scroll STATIC
- for
- select a.makh, tenkh from KHACHHANG a,HOADON b
- where a.MAKH=b.MAKH and month(NGAY)=@t and
- year(NGAY)=@n
- open @cur

- go

- DECLARE @mcur CURSOR
- EXEC P6 6,2000, @cur = @mcur OUTPUT
- FETCH first FROM @mcur
- WHILE (@@FETCH_STATUS = 0)
- BEGIN
- FETCH NEXT FROM @mcur
- END
- CLOSE @mcur
- DEALLOCATE @mcur

Trong NorthWind

- Khai báo một procedure CustomersOfCountryCursor để lấy ra một cursor chứa các record của table customers có country bằng giá trị truyền vào. Thực thi CustomersOfCountryCursor và in ra các dữ liệu có trong cursor trả về.

- Truyền vào năm và trả về doanh thu theo tháng trong năm đó
- Giải
- CREATE PROC P4 @nam integer,
- @Cur CURSOR VARYING OUTPUT
- AS
- SET @Cur = CURSOR scroll STATIC
- FOR
- SELECT month(ngay) as thang, sum(sl*giaban) as doanhthu
- FROM HD, CTHD
- where HD.mahd = CTHD.mahd and
- year(ngay)=@nam