# MỤC TIÊU

* Trong csdl có thể có nhiều đối tượng như table, views, thủ tục,trigger,..và hàm cũng là 1 đối tượng trong csdl. Ý tưởng chính chủ yếu là tránh việc lặp đi lặp lại nhiều lần cùng 1 đoạn mã, giúp ta kiểm soát đầu vào là gì, cũng như cấu trúc, loại đầu ra, ai có thể sử dụng chúng và người đó sẽ làm theo cách nào. Hàm là 1 công cụ rất mạnh hỗ trợ cho chúng ta quản trị tốt csdl.
* Ví dụ bạn cần viết một loạt các câu lệnh SQL phức tạp lên đến hàng ngàn dòng thì có thể phân tích và chia nhỏ thành nhiều hàm.
* Cũng như thủ tục, việc sử dụng hàm trong SQL Server giúp đơn giãn hóa công việc lập trình bằng cách đóng gói các lệnh SQL thành 1 nhóm để có thể dễ mở rộng, tái sử dụng nhiều lần, rút ngắn thời gian thực hiện công việc.
* Mục tiêu buổi học hôm nay sẽ đi tìm hiểu tổng quan về hàm người dùng định nghĩa. Sau buổi học chúng ta sẽ phân biệt được hàm giá trị vô hướng và hàm giá trị bảng. Cũng như biết cách TẠO,gọi, thay đổi và xóa hàm.

# SLIDE 5: HÀM NGƯỜI DÙNG ĐỊNH NGHĨA

* Là một đối tượng CSDL chứa các câu lệnh SQL, được biên dịch sẵn và lưu trữ trong CSDL.
* Thực hiện một hành động như các tính toán phức tạp và trả về kết quả là một giá trị.
* Giá trị trả về có thể là: Giá trị vô hướng (HÀM VÔ HƯỚNG) hoặc Một bảng (HÀM BẢNG)

# SLIDE 6: PHÂN BIỆT HÀM và THỦ TỤC

* GN: Cả 2 đều là đối tượng trong csdl chứa các câu lệnh SQL biên dịch sẵn và lưu trong csdl
* KN:
* HÀM LUÔN trả về 1 giá trị (là giá trị đơn hoặc tập kq dạng bảng) dưới lệnh RETURN. CÒN THỦ TỤC có thể trả về giá trị hoặc không.
* HÀM KHÔNG có tham số đầu ra, THỦ TỤC có thể có hoặc không
* HÀM KHÔNG chứa các câu lệnh INSERT, UPDATE, DELETE 1 bảng hoặc view tồn tại trong csdl. THỦ TỤC thì có thể
* Trong thân HÀM, ta có thể tạo bảng, bảng tạm, biến bảng và thực thi các câu lệnh INSERT, UPDATE, DELETE trên các bảng này

# SILDE 7: VD HÀM TÍNH TUỔI

* Ta sử dụng hàm OBJECT\_ID: để kiểm tra sự tồn tại của 1 đối tượng trong csdl. Cú pháp:

OBJECT\_ID('TENDOITUONG','DẠNGDOITUONG')

Xem vd

# SLIDE 9: CÁC LOẠI HÀM NGƯỜI DÙNG ĐỊNH NGHĨA

Đọc slide

# SLIDE 10: TẠO HÀM GIÁ TRỊ VÔ HƯỚNG

* Cú pháp tạo hàm khá tương tự với tạo 1 thủ tục.

CREATE FUNCTION tenluocdo.tenham (@thamsovao kieudulieu [=giatrimacdinh])

RETURNS kieugiatritrave

AS

BEGIN

--LENHSQL

RETURN bieuthucvohuong

END

* Phần CREATE FUNCTION nếu k chỉ định tên lược đồ schema cho hàm sẽ mặc định tạo hàm trên schema liên kết với user đang đăng nhập.
* Phần danh sách tham số: Cấu trúc tương tự Stored Procedure. Nhưng không có tham số đầu ra
* Phần thân hàm. Phải được chứa trong câu lệnh BEGIN…END. Và phải có hàm trả về giá trị tương ứng với kiểu dữ liệu trả về được khai báo.
* Không thể truyền tham số theo tên
* Khi gọi hàm bắt buộc phải truyền đầy đủ các tham số theo vị trí. Kể cả tham số tùy chọn, nếu muốn sử dụng giá trị mặc định, phải đặt từ khóa DEFAULT tại đúng vị trí tham số tùy chọn đó.

# SLIDE 11: VD TẠO HÀM VÔ HƯỚNG KHÔNG CÓ THAM SỐ

* Hàm vô hướng sẽ trả về giá trị vô hướng, tức 1 giá trị đơn.
* Ví dụ ta tạo hàm đếm tổng số nhân viên. Hàm này không có tham số đầu vào và trả về số lượng nv là 1 giá trị vô hướng,

CREATE FUNCTION F\_DEMNV ()

RETURNS int

BEGIN

RETURN(

SELECT COUNT(MANV)

FROM NHANVIEN

)

END

* Để gọi hàm vô hướng các bạn dùng lệnh SELECT hoặc PRINT. Khác với thủ tục, đối với các tham số đầu vào tùy chọn (tức có giá trị mặc định) ta vẫn phải truyền giá trị cho nó. Nếu muốn sử dụng giá trị mặc định ta dùng DEFAULT

SELECT DBO.fDemNv() hoặc

PRINT N'Tổng số nhân viên là: '+CONVERT(NVARCHAR,DBO.F\_DEMNV())

* Để xem lại các hàm vừa tạo, trong csdl QLDA, tại thư mục FUNCTIONS các bạn thấy có 4 loại hàm:
* TABLE VALUED FUNCTIONS (HÀM BẢNG tức giá trị trả về dạng bảng)
* SCALAR VALUED FUNCTIONS (HÀM VÔ HƯỚNG tức giá trị trả về dạng giá trị vô hướng)
* AGGREGATE FUNCTIONS (HÀM TẬP HỢP là các hàm toán học tổng cộng một tập hợp các giá trị để cho một giá trị kết quả duy nhất. VD: MAX, MIN, AVG,COUNT,..
* SYSTEM FUNCTIONS: HÀM HỆ THỐNG
* Các bạn vào SCALAR VALUED FUNCTIONS sẽ thấy hàm vô hướng mà chúng ta vừa tạo.

# SLIDE 12: VD2 HÀM VÔ HƯỚNG CÓ THAM SỐ

--Tạo hàm nhận tham số đầu vào là giới tính nam hoặc nữ và

--đếm số lượng nhân viên theo giới tính.

ALTER FUNCTION F\_NVPHAI (@phai nvarchar(3))

RETURNS INT

AS

BEGIN

RETURN (

SELECT COUNT(MANV)

FROM NHANVIEN

WHERE PHAI LIKE @phai

)

END

SELECT DBO.F\_NVPHAI('nam')

# SLIDE 13: HÀM BẢNG

* Hàm giá trị bảng đơn giản còn gọi là hàm giá trị bảng nội tuyến(tức bảng này nằm trong csdl của chúng ta)
* Hàm giá trị bảng đơn giản có thể được dùng trong câu lệnh truy vấn thay thế cho tên bảng hoặc tên view

CREATE FUNCTION tenham (@thamsovao kieudulieu [=giatrimacdinh])

RETURNS TABLE

AS

RETURN (

Lenh sql

)

Gọi hàm

SELECT \* FROM tenham(giatrithamso)

# SLIDE 14: VD HÀM BẢNG

--Viết hàm trả về bảng các nhân viên làm việc ở phòng số 5

CREATE FUNCTION F\_NV (@ma int)

RETURNS TABLE

AS

RETURN (

SELECT \*

FROM NHANVIEN

WHERE PHG =@ma

)

SELECT \* FROM F\_NV(1)

# SLIDE 15: HÀM BẢNG ĐA CÂU LỆNH

CREATE FUNCTION TENHAM (@TENTHAMSO KIEUDULIEU [=GIATRI])

RETURNS @TENBANG TABLE (

TENCOT KIEUDULIEU

)

AS

BEGIN

LENH SQL

RETURN

END

Thực thi hàm

SELECT \* FROM tenham (giatrithamso)

# SLIDE 16: VD HÀM BẢNG ĐA CÂU LỆNH

--Viết hàm nhận giá trị từ phòng ban, tham số đầu vào là mã phòng, nếu truyền NULL

--thì hàm nhận tất cả các dòng từ bảng PHONGBAN

CREATE FUNCTION F\_PB (@ma int)

RETURNS @A TABLE (

TEN NVARCHAR(15),

MA INT,

TRUONG NVARCHAR(9),

NGAY DATE

)

AS

BEGIN

IF @ma IS NULL

BEGIN

INSERT INTO @A

SELECT \* FROM PHONGBAN

END

ELSE

BEGIN

INSERT INTO @A

SELECT \*

FROM PHONGBAN

WHERE MAPHG=@ma

END

RETURN

END

SELECT \* FROM F\_PB (NULL)

# SLIDE 17: XÓA,SỬA HÀM

Để xóa hàm DROP FUNCTION TENHAM

Để sửa hàm vô hướng ALTER FUNCTION

# SLIDE 18 DEMO

--Tạo hàm nhận tham số đầu vào là mã phòng, trả về bảng các nhân viên

--thuộc mã phòng đó.

CREATE FUNCTION F\_NV (@ma int)

RETURNS TABLE

AS

RETURN (

SELECT \*

FROM NHANVIEN

WHERE PHG=@ma

)

SELECT \* FROM F\_NV(1)

# LƯU Ý

HÀM VÔ HƯỚNG: khi thực thi bắt buộc có DBO.TENHAM

HÀM BẢNG ĐƠN GIẢN: KHÔNG CÓ BEGIN END sau AS

HÀM BẢNG ĐA CÂU LỆNH: Lệnh RETURN ở cuối thân hàm chứa trong BEGIN END

# LAB 7 B1

--Nhập vào MaNV cho biết tuổi của nhân viên này.

IF OBJECT\_ID ('F\_TUOI','FN') IS NOT NULL --FN: Hàm vô hướng

DROP FUNCTION F\_TUOI

GO

CREATE FUNCTION F\_TUOI (@ma nvarchar(9))

RETURNS INT

AS

BEGIN

RETURN (

SELECT YEAR(GETDATE())-YEAR(NGSINH)

FROM NHANVIEN

WHERE MANV=@ma

)

END

GO

PRINT N'Tuổi của nhân viên là: '+CONVERT(NVARCHAR,DBO.F\_TUOI('001'))

SELECT \* FROM NHANVIEN WHERE MANV='001'

--➢ Nhập vào Manv cho biết số lượng đề án nhân viên này đã tham gia

IF OBJECT\_ID ('F\_SLDEAN','FN') IS NOT NULL

DROP FUNCTION F\_SLDEAN

GO

CREATE FUNCTION F\_SLDEAN (@ma nvarchar(9))

RETURNS INT

AS

BEGIN

RETURN (

SELECT COUNT(MADA)

FROM PHANCONG

WHERE MA\_NVIEN=@ma

)

END

GO

PRINT N'Số lượng đề án nhân viên đã tham gia là: '+ CONVERT(NVARCHAR,DBO.F\_SLDEAN('001'))

SELECT \* FROM PHANCONG WHERE MA\_NVIEN='001'

--➢ Truyền tham số vào phái nam hoặc nữ,

--xuất số lượng nhân viên theo phái

IF OBJECT\_ID('F\_SL\_PHAI','FN') IS NOT NULL

DROP FUNCTION F\_SL\_PHAI

GO

CREATE FUNCTION F\_SL\_PHAI (@phai nvarchar(3))

RETURNS INT

AS

BEGIN

RETURN (

SELECT COUNT(MANV)

FROM NHANVIEN

WHERE PHAI LIKE @phai

)

END

GO

PRINT N'Số lượng nhân viên là: '+ CONVERT(NVARCHAR,DBO.F\_SL\_PHAI('Nam'))

SELECT \* FROM NHANVIEN WHERE PHAI LIKE 'Nam'

-- Truyền tham số đầu vào là tên phòng, tính mức lương trung bình

--của phòng đó, Cho biết họ tên nhân viên (HONV, TENLOT, TENNV)

--có mức lương trên mức lương trung bình của phòng đó.

IF OBJECT\_ID('F\_NV\_LUONG','FT') IS NOT NULL

DROP FUNCTION F\_NV\_LUONG

GO

CREATE FUNCTION dbo.F\_NV\_LUONG (@ten nvarchar(30))

RETURNS @nv\_luong TABLE (

HONV NVARCHAR(15),

TENLOT NVARCHAR(15),

TENNV NVARCHAR(15)

)

AS

BEGIN

INSERT INTO @nv\_luong

SELECT HONV,TENLOT,TENNV

FROM NHANVIEN

WHERE LUONG > (

SELECT AVG(LUONG)

FROM PHONGBAN a, NHANVIEN b

WHERE TENPHG LIKE @ten

AND a.MAPHG=b.PHG

)

RETURN

END

SELECT \* FROM DBO.F\_NV\_LUONG(N'Điều hành')

--Tryền tham số đầu vào là Mã Phòng, cho biết tên phòng ban,

--họ tên người trưởng phòng và số lượng đề án mà phòng ban đó chủ trì.

IF OBJECT\_ID ('F\_PHONGBAN','FT') IS NOT NULL

DROP FUNCTION F\_PHONGBAN

GO

CREATE FUNCTION F\_PHONGBAN (@ma int)

RETURNS @A TABLE (

TENPB nvarchar(30),

TENTR nvarchar (30),

SODEAN int

)

AS

BEGIN

INSERT INTO @A

SELECT a.TENPHG,c.HONV+ ' '+c.TENLOT+ ' '+c.TENNV,COUNT(b.MADA)

FROM PHONGBAN a, DEAN b, NHANVIEN c

WHERE a.MAPHG=b.PHONG

AND a.TRPHG=c.MANV

AND MAPHG=@ma

GROUP BY a.TENPHG,c.HONV+ ' '+c.TENLOT+ ' '+c.TENNV

RETURN

END

go

SELECT \* FROM F\_PHONGBAN(4)