# MỤC TIÊU

* Trong những bài học trước đây khi dùng Query chúng ta có thể save các nhóm câu lệnh SQL vào một file dưới dạng script để có thể sử dụng trở lại sau này. Tuy nhiên thay vì save vào text file ta có thể save vào trong SQL Server dưới dạng Stored Procedure hay gọi tắt là SP
* Có khá nhiều lợi ích khi chúng ta sử dụng SP
* Ví dụ chúng ta có một thao tác như sau: Kiểm tra số lượng thân nhân của nhân viên A. Công việc này được lặp đi lặp lại khá nhiều lần. Thay vì viết nhiều dòng mã T-SQL, chúng ta sẽ đóng gói chúng lại thành 1 procedure. Mỗi khi cần, chúng ta gọi thực thi nó. Việc tái sử dụng code giúp công việc nhanh và đơn giản hơn rất nhiều so với việc viết lại câu truy vấn.
* Xét về phương thức hoạt động, ta có thể hình dung SP sẽ được biên dịch vào lần chạy đầu tiên, sau đó nó sẽ được lưu vào bộ nhớ tạm (cache). Những lần chạy sau sẽ nhanh hơn và tiết kiệm tài nguyên của hệ thống.
* Sử dụng SP, Chúng ta chỉ cần thực hiện 1 câu lệnh (gọi hàm thực thi), thay vì vài thực thi lần lượt hàng loạt câu lệnh. Việc này giúp việc giảm giao tiếp liên tục giữa client và server, giảm thiểu các rủi ro do đường truyền gây ra.
* Mục tiêu của buổi học hôm nay chúng ta sẽ đi tìm hiểu tổng quan về thủ tục SQL STORED PROCEDURES. Nắm được các khái niệm, biết cách tạo và thực thi một SP.

# SLIDE 5 STORED PROCEDURES

* Khi học lập trình C, ta hay dùng khái niệm chương trình con, thì 1 thủ tục hoàn chỉnh Stored Procedure có thể hiểu là 1 chương trình con.
* Stored Procedure gọi tắt là SP, là một nhóm câu lệnh Transact-SQL đã được compiled (biên dịch) và chứa trong SQL Server dưới một tên nào đó và được xử lý như một đơn vị (tương tự với table,views,…) (chứ không phải nhiều câu SQL riêng lẻ).
* Ngay khi tạo thủ tục nó đã được biên dịch thành dạng p-code vì thế nâng cao khả năng thực hiện, sẵn sàng cho tái sử dụng khi cần, nhằm làm tăng tốc độ xử lý, khả năng sử dụng chung, tăng tính bảo mật an toàn dữ liệu và tăng tiện ích trong phát triển.
* Các bạn nhìn vào mô hình, LINQ hỗ trợ tạo 1 thủ tục với tên ngắn gọn, khi lập trình chỉ cần gọi tên đó trong phần code C#, visual basic,..nó sẽ kết nối với csdl gọi thủ tục đó lên thực thi câu lệnh SQL, như vậy khi lập trình k cần viết câu lệnh INSERT, UPDATE, DELETE mà chỉ cần gọi đúng tên thủ tục, tự động thủ tục kết nối với csdl để thực hiện câu lệnh. Như vậy làm cho phần lập trình không phụ thuộc phần code liên quan csdl, code csdl được lưu trữ trong thủ tục nằm trong hệ quản trị. Từ đó, đảm bảo tính độc lập hệ quản trị, sau này khi hệ quản trị dù có thay đổi cú pháp cũng không ảnh hưởng, ta chỉ cần vào trong thủ tục cập nhật lại là xong.

# SLIDE 6 ĐẶC TÍNH CỦA SP

* Đặc tính của Stored-procedure trong SQL Server
* Chấp nhận những tham số vào và trả về những giá trị được chứa trong các tham số ra để gọi những thủ tục hoặc xử lý theo lô.
* Chứa các lệnh của chương trình để thực hiện các xử lý trong database, bao gồm cả lệnh gọi các thủ tục khác thực thi
* Trả về các trạng thái giá trị để gọi những thủ tục hoặc thực hiện các xử lý theo lô để cho biết việc thực hiện thành công hay thất bại, nếu thất bại thì lý do vì sao thất bại.

# SLIDE 7 THUẬN LỢI KHI SỬ DỤNG SP

Lợi ích của Stored Procedures để làm gì?

* Thứ nhất đó là đảm bảo tính module hóa: nghĩa là chúng ta chỉ viết SP 1 lần, sau đó ta có thể gọi để sử dụng nhiều lần trong ứng dụng.
* Thứ 2 là về hiệu suất: SP giúp thực thi mã nhanh hơn cách viết từng câu lệnh sql thông thường, SP đã được biên dịch và lưu sẵn tại SQL server, do vậy chương trình không cần gởi hàng ngàn dòng lệnh đến SQL Server. Nó chỉ cần gọi một dòng lệnh để thực thi SP, điều này giúp tăng tốc độ thực thi.
* SP có thể làm giảm bớt vấn đề kẹt đường truyền mạng, dữ liệu được gởi theo gói. Bình thường khi ta gởi nhiều câu lệnh SQL thông qua đường truyền mạng thì có thể ảnh hưởng đến hiệu suất đường truyền. Khi dùng SP ta sẽ gom các câu lệnh vào cùng 1 SP và chỉ gọi 1 lần duy nhất qua đường truyền mạng nên rõ ràng sẽ giảm tải được băng thông.
* Cuối cùng là tính bảo mật. SP làm cho người dùng chỉ truy cập gián tiếp đến các tác vụ cấp cao nên k làm ảnh hưởng đến tính bảo mật của SQL Server.
* Có 2 loại Store Procedure chính: thủ thục hệ thống System stored procedures và thủ tục người dùng định nghĩa User stored procedures

# SLIDE 8: CÁC THÀNH PHẦN CỦA SP

* Inputs: nhận các tham số đầu vào khi cần
* Execution: kết hợp giữa các yêu cầu nghiệp vụ với các lệnh lập trình như IF..ELSE, WHILE...
* Outputs: trả ra các đơn giá trị (số, chuỗi…) hoặc một tập kết quả.
* Bên trái ta dùng câu lệnh SQL thông thường.
* Bên phải ta tạo SP tên SearchClientByState, input nhận tham số đầu vào là State kiểu varchar. Khối lệnh là các câu lệnh SQL tương tự bên trái. Để thực thi lệnh ta dùng EXECUTE tênthutuc và giá trị truyền cho tham số đầu vào. Đầu ra ta nhận được là tập kết quả họ, tên, số điện thoại của khách hàng có trạng thái trùng với giá trị tham số mà ta truyền vào.

# SLIDE 9: VD CỤ THỂ CÁC THÀNH PHẦN CỦA SP

* Để hình dung rõ hơn các thành phần trong SP, ta xét vd sau đây:
* Đầu tiên ta tạo thủ tục tên …output có 1 tham số đầu vào là …kiểu int
* Bắt đầu lệnh SP bằng từ khóa AS
* PhẦn Execution chứa câu lệnh truy vấn SELECT
* PhẦn Output trả về 1 dòng kết quả thỏa điều kiện ở mệnh đề WHERE là …=giá trị tham số ta truyền vào
* Thông thường T-SQL sẽ thông báo tổng số các bản ghi bị tác động khi thực thi một lệnh SQL. Tuy nhiên, nếu chúng ta sử dụng lệnh SET NOCOUNT ON ở đầu các stored procedures thì hệ thống không trả về thông báo nữa. Việc này tiết kiệm lưu lượng mạng chút xíu.

# SLIDE 10: TẠO SP

* **Cú pháp để tạo 1 SP như sau:**

CREATE PROCEDURE tenthutuc --HOẶC CREATE PROC

@tenbien1 kieudulieU, --bienvao

@tenbien2 kieudulieu OUT --bienra

AS

BEGIN

[DECLARE bienkq kieudulieu]

--khai bao cac bien cho xu ly

--cac cau lenh SQL

[RETURN @bienkq] --KHI CÓ GIÁ TRỊ TRẢ VỀ

END

* Tên hàm, tên biến trong SQL Server không phân biệt hoa thường.
* Có thể thay thế **Create Procedure** bằng **Create Proc**
* Tham số đầu vào
* Tham số bắt buộc: bắt buộc phải truyền
* Tham số tuỳ chọn: gán giá trị mặc định, giá trị mặc định nếu không gọi.
* **Cú pháp thực thi một SP**
* Khi SP không có tham số vào

EXECUTE TENSP –hoac EXEC

* Khi SP có tham số vào

DECLARE @biendaura

EXECUTE TENSP giatri1 [,giatri 2,...,@biendaura OUT]

* Khi SP có giá trị trả về

DECLARE @bienkq int

EXECUTE @bienkq= TENTHUTUC 'GIATRITRUYENVAO' –Hoac EXEC

SELECT @bienkq

# SLIDE 11: VD TẠO THỦ TỤC TÍNH TỔNG 2 SỐ NGUYÊN

--Tính tổng 2 số a b

CREATE PROCEDURE SP\_TONG @a int, @b int

AS

BEGIN

DECLARE @tong int

SET @tong=@a+@b

PRINT @tong

END

--Thực thi EXECUTE SP\_TONG 4,5

--Tính tổng 2 số a b có đầu ra

CREATE PROCEDURE SP\_TONG1 @a int, @b int, @tong int **OUT**

AS

BEGIN

SET @tong=@a+@b

END

--Thực thi

DECLARE @tong int

EXECUTE SP\_TONG 4,5,@tong OUT

SELECT @tong

# SLIDE 12: TẠO SP VÀ THỰC THI SP CÓ GIÁ TRỊ TRẢ VỀ

* Khi tạo SP có giá trị trả về ta thêm lệnh

RETURN @bienkq

* Để gọi SP:

DECLARE @bienkq int

EXECUTE @bienkq= TENTHUTUC 'GIATRITRUYENVAO' –Hoac EXEC

SELECT @bienkq

--Tạo thủ tục có giá trị trả về là số lượng NV ở HCM

CREATE PROC SP\_NVHCM @tp nvarchar(30)

AS

BEGIN

DECLARE @soluong int

SELECT @soluong=COUNT(\*) FROM NHANVIEN

WHERE DCHI LIKE N'%'+@tp

RETURN @soluong

END

--Thực thi SP

DECLARE @sl int

EXECUTE @sl=SP\_NVHCM 'HCM'

SELECT @sl

# SLIDE 14 TẠO SP CÓ THAM SỐ MẶC ĐỊNH

Những trường hợp này nếu khi thực thi SP mà ta không truyền tham số cho SP thì nó sẽ sử dụng tham số mặc định đề thực thi.

CREATE PROC SP\_TONG1 @a int=1, @b int=2

AS

BEGIN

DECLARE @tong int

SET @tong=@a+@b

PRINT @tong

END

--Không truyền tham số dùng giá trị mặc định

EXEC SP\_TONG1 =>3

--Khi truyền tham số

EXEC SP\_TONG1 3,5 =>8

# SLIDE 15:VD TRUYỂN THAM SỐ LÀ MANV, HIỂN THỊ THÔNG TIN NV

--Tạo SP hiển thị nhân viên khi truyền tham số manv

CREATE PROC SP\_NV @manv varchar(10)

AS

BEGIN

SELECT \* FROM NHANVIEN WHERE MANV=@manv

END

EXEC SP\_NV '001'

--Tạo SP với tham số đầu vào là mã phòng,tên phòng,

--trưởng phòng, ngày nhận chức

--thực hiện thêm dữ liệu vào bảng PHONGBAN nếu mã phòng chưa tồn tại

--update PHONGBAN nếu mã phòng đã tồn tại

--LƯU Ý: TRUYỀN ĐÚNG THỨ TỰ

CREATE PROC SP\_ThemPB @tenphong nvarchar(30),@maphong int,

@truongphong nvarchar(30), @ngay date

AS

BEGIN

IF EXISTS (SELECT \* FROM PHONGBAN WHERE MAPHG=@maphong)

UPDATE PHONGBAN

SET TENPHG=@tenphong, TRPHG=@truongphong,NG\_NHANCHUC=@ngay

WHERE MAPHG=@maphong

ELSE

INSERT INTO PHONGBAN VALUES

(@tenphong,@maphong,@truongphong,@ngay)

END

EXEC SP\_ThemPB N'Nhân sự',7,'005','05-26-2022'

# SLIDE 16 VD

--Viết store procedure nhận vào tham số là năm sinh,

--xuất ra tên các nhân viên.

CREATE PROC SP\_TENNV @ns int

AS

BEGIN

SELECT TENNV

FROM NHANVIEN

WHERE YEAR(NGSINH)=@ns

END

GO

EXEC SP\_TENNV 1984

--Viết store procedure đếm số lượng thân nhân

--của nhân viên có mã nhân viên được nhập từ người dùng

CREATE PROC SP\_DEMTN @msnv nvarchar(9)

AS

BEGIN

SELECT COUNT(TENTN) AS N'SỐ THÂN NHÂN'

FROM NHANVIEN a, THANNHAN b

WHERE a.MANV=b.MA\_NVIEN

AND MA\_NVIEN=@msnv

END

GO

EXEC SP\_DEMTN '009'

# LAB5 B1

-- In ra dòng ‘Xin chào’ + @ten với @ten là tham số đầu vào là tên Tiếng Việt có dấu

CREATE PROC SP\_INTEN @ten nvarchar(20)

AS

PRINT N'Xin chào ' + @ten

GO

EXEC SP\_INTEN N'Lan Vy'

--Nhập vào 2 số @s1,@s2. In ra câu ‘Tổng là : @tg’ với @tg=@s1+@s2.

CREATE PROC SP\_TONG2SO @s1 int, @s2 int, @tg int OUT

AS

BEGIN

SET @tg=@s1+@s2

PRINT N'Tổng là: '+ CONVERT(NVARCHAR,@tg)

END

GO

DECLARE @tong int

EXEC SP\_TONG2SO 20,10, @tong OUT

--Nhập vào số nguyên @n. In ra tổng các số chẵn từ 1 đến @n.

CREATE PROC SP\_TONGNSO @n int

AS

BEGIN

DECLARE @i int=1,@tong int=0

WHILE @i<=@n

BEGIN

IF (@i%2=0)

BEGIN

SET @tong=@tong+@i

SET @i=@i+1

END

ELSE SET @i=@i+1

END

PRINT N'Tổng các số chẵn từ 1 đến '+ CONVERT(NVARCHAR,@n) +

N' là: '+ CONVERT(NVARCHAR,@tong)

END

GO

EXEC SP\_TONGNSO 10

--Nhập vào 2 số. In ra ước chung lớn nhất của chúng theo gợi ý

-- b1. Không mất tính tổng quát giả sử a <= A

--Nếu A%a=0 thì : (a,A) = a, ngược lại : (a,A) = (A%a,a) hoặc (a,A) =(a,A-a)

--b3. Lặp lại b1,b2 cho đến khi điều kiện trong b2 được thỏa

ALTER PROC SP\_UCLN @a int, @a1 int

AS

BEGIN

DECLARE @ucln int, @b int=@a, @b1 int=@a1

WHILE @a<=@a1

BEGIN

IF @a1%@a=0

BEGIN

SET @ucln=@a;

PRINT N'UCLN của '+CONVERT(NVARCHAR,@b)+ ' và '+

CONVERT(NVARCHAR,@b1)+ ' là: '+ CONVERT(NVARCHAR,@ucln)

BREAK;

END

ELSE

BEGIN

SET @a1=@a1%@a

IF @a1<@a

BEGIN

DECLARE @tam int

SET @tam=@a

SET @a=@a1

SET @a1=@tam

END

END

END

END

GO

EXEC SP\_UCLN 93,333

# LAB 5 BÀI 2

--Nhập vào @Manv, xuất thông tin các nhân viên theo @Manv.

CREATE PROC SP\_NV @manv varchar(10)

AS

BEGIN

SELECT \* FROM NHANVIEN WHERE MANV=@manv

END

GO

EXEC SP\_NV '001'

--Nhập vào @MaDa (mã đề án), cho biết số lượng nhân viên tham gia đề án đó

CREATE PROC SP\_DEMNV\_DEAN @mada int, @sl int OUT

AS

BEGIN

SELECT @sl=COUNT(MA\_NVIEN)

FROM PHANCONG

WHERE MADA=@mada

PRINT N'Số lượng nhân viên tham gia đề án '+CONVERT(nvarchar,@mada)

+ ' là: '+CONVERT(NVARCHAR,@sl)

END

GO

DECLARE @sl int

EXEC SP\_DEMNV\_DEAN 1,@sl OUT

--ALTER TABLE PHONGBAN ADD UNIQUE(TRPHG)

--Nhập vào @Trphg (mã trưởng phòng), xuất thông tin các nhân viên có trưởng phòng là @Trphg và các nhân viên này không có thân nhân.

CREATE PROC SP\_NV\_TRPHG @trphg nvarchar(20)

AS

BEGIN

SELECT \*

FROM NHANVIEN

WHERE PHG =(

SELECT MAPHG

FROM PHONGBAN

WHERE TRPHG=@trphg

)

AND MANV NOT IN (

SELECT MA\_NVIEN

FROM THANNHAN

)

END

GO

EXEC SP\_NV\_TRPHG '006'

--Nhập vào @Manv và @Mapb, kiểm tra nhân viên có mã @Manv có thuộc phòng ban có mã @Mapb hay không

CREATE PROC SP\_KTNVPB @Manv nvarchar(9), @Mapb int

AS

IF EXISTS ( SELECT \*

FROM NHANVIEN

WHERE PHG=@Mapb

AND MANV=@Manv)

PRINT N'Nhân viên '+CONVERT(nvarchar, @Manv) +

N' thuộc phòng '+CONVERT(nvarchar, @Mapb)

ELSE

PRINT N'Nhân viên '+CONVERT(nvarchar, @Manv) +

N' không thuộc phòng '+CONVERT(nvarchar, @Mapb)

GO

EXEC SP\_KTNVPB '001',5

--Nhập vào @MaDa và @Ddiem\_DA (địa điểm đề án), cho biết số lượng nhân viên tham gia đề án có mã đề án là @MaDa và địa điểm đề án là @Ddiem\_DA

CREATE PROC SP\_DEMNV\_DADD @MaDa int, @Ddiem\_DA nvarchar(40)

AS

SELECT COUNT(MA\_NVIEN) AS N'SỐ LƯỢNG NV THAM GIA ĐỀ ÁN'

FROM PHANCONG a, CONGVIEC b,DEAN c

WHERE a.MADA=b.MADA

AND a.STT=b.STT

AND b.MADA=c.MADA

AND a.MADA=@MaDa

AND DDIEM\_DA=@Ddiem\_DA

GO

EXEC SP\_DEMNV\_DADD 1, N'Vũng Tàu'