

## BÀI TẬP 2

- SV tái tạo (restore database trong SQL Server) CSDL **QLNhanVien** từ file “**QLNhanVien.bak**” (đề cho) để làm bài.
- SV thực hiện tất cả các câu đều phải có đủ 2 phần: **tạo** stored procedure và **test** stored procedure
- SV lưu file bài làm vào ổ **S:\SoMay\_MSSV\_HoTen.sql**.

Cho CSDL **quản lý nhân viên** có 2 bảng như sau:

NhanVien			
	Column Name	Data Type	Allow Nulls
	MaNV	int	<input type="checkbox"/>
	Ho	nvarchar(50)	<input checked="" type="checkbox"/>
	Ten	nvarchar(20)	<input checked="" type="checkbox"/>
	MaPB	nvarchar(10)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

PhongBan			
	Column Name	Data Type	Allow Nulls
	MaPB	nvarchar(10)	<input type="checkbox"/>
	TenPB	nvarchar(50)	<input checked="" type="checkbox"/>
	ChiNhanh	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

**NhanVien**(MãNV, Họ, Tên, MaPB)

**PhòngBan**(MãPB, TênPB, ChiNhanh)

Khóa chính: là MãNV và MãPB đã được gạch dưới

Khóa ngoại: **NhanVien. MaPB** là khóa ngoại tham khảo đến **PhòngBan. MãPB**

Dữ liệu từ 2 bảng hiện có:

MaPB	TenPB	ChiNhanh
PB01	Thiết kế	Sài gòn
PB02	Bán hàng	Sài gòn
PB03	Tiếp thị	Sài gòn
PB04	Sản xuất	Hà nội
PB05	Kinh doanh	Hà nội
PB06	Kế toán	Hà nội

MaNV	Ho	Ten	MaPB
1	Nguyễn Văn	Trung	PB01
2	Hồ Thanh	Tùng	PB02
3	Lê Ánh	Hồng	PB03
4	Trần Thị	Thắm	PB01
5	Đặng Trung	Thành	PB02
6	Mai Ngọc	Hòa	PB03
7	Phạm Văn	Minh	PB04
8	Nguyễn Thanh	Thảo	PB05
9	Trần Văn	Quang	PB06
10	Bành Văn	Thắng	PB04
11	Đặng Ngọc Thu	Hà	PB05
12	Lê Văn	Mạnh	PB06

Dữ liệu kết từ 2 bảng hiện có:

MaNV	Ho	Ten	MaPB	TenPB	ChiNhanh
1	Nguyễn Văn	Trung	PB01	Thiết kế	Sài gòn
2	Hồ Thanh	Tùng	PB02	Bán hàng	Sài gòn
3	Lê Ánh	Hồng	PB03	Tiếp thị	Sài gòn
4	Trần Thị	Thắm	PB01	Thiết kế	Sài gòn
5	Đặng Trung	Thành	PB02	Bán hàng	Sài gòn
6	Mai Ngọc	Hòa	PB03	Tiếp thị	Sài gòn
7	Phạm Văn	Minh	PB04	Sản xuất	Hà nội
8	Nguyễn Thanh	Thảo	PB05	Kinh doanh	Hà nội
9	Trần Văn	Quang	PB06	Kế toán	Hà nội
10	Bành Văn	Thắng	PB04	Sản xuất	Hà nội
11	Đặng Ngọc Thu	Hà	PB05	Kinh doanh	Hà nội
12	Lê Văn	Mạnh	PB06	Kế toán	Hà nội

**Câu 1:** Hãy viết code SQL để tạo 2 Stored procedure tên **TaoPhanManhPB** và **TaoPhanManhNV** để tạo và lấy dữ liệu cho các phân mảnh của 2 bảng PhòngBan và NhânViên, biết:

- Bảng PhòngBan được phân mảnh ngang chính dựa vào cột ChiNhánh. Biết cột ChiNhánh chỉ có đúng 2 giá trị là “Sài gòn” và “Hà nội”.
- Bảng NhânViên được phân mảnh ngang dẫn xuất theo các phân mảnh của bảng PhòngBan.

**Câu 2, 3, 4 và 5:** Sinh viên hãy viết code SQL để tạo 2 Stored procedure cho mỗi câu sau đây ở mức 1 (fragmentation transparency - trong suốt phân mảnh) và mức 2 (location transparency – trong suốt vị trí):

**Câu 2:** Viết code SQL để tạo 2 stored procedure tên **DSNhanVienMuc1** và **DSNhanVienMuc2** để lập danh nhân viên thuộc các phòng ban biết tên phòng ban. Danh nhân viên gồm 6 cột: **MãNV**, **Họ, Tên**, **MãPB**, **TênPB**, **ChiNhánh**, tham số vào của 2 stored procedure là @TênPB.

Yêu cầu: phải có báo lỗi chi tiết (print ra màn hình) khi gặp các trường hợp ngoại lệ sau:

- @TênPB chứa giá trị NULL
- @TênPB chứa giá trị không tìm thấy trong CSDL

Sau đó dùng 2 stored procedure này để TEST lập danh nhân viên biết tên phòng ban:

Giá trị tham số @TênPB	Trường hợp thử	Kết quả cần có
(N'Thiết kế')	trường hợp có tìm thấy tên PB	→ hiện kết quả
(N'Kế toán')	trường hợp có tìm thấy tên PB	→ hiện kết quả
(N'Kỹ thuật')	trường hợp không tìm thấy tên PB	→ hiện báo lỗi “Không tìm thấy tên phòng ban!”
(NULL)	trường hợp không truyền vào tên PB	→ hiện báo lỗi “Không nhập tên phòng ban!”

**Câu 3:** Viết code SQL để tạo 2 stored procedure tên **ThemPhongBanMuc1** và **ThemPhongBanMuc2** để thêm 1 hàng dữ liệu. Các tham số vào là @MãPB, @TênPB, và @ChiNhánh.

Yêu cầu: phải có báo thành công khi thêm dữ liệu thành công, hay báo lỗi chi tiết (print ra màn hình) và không thêm dữ liệu khi gặp các trường hợp ngoại lệ sau:

- Có 1, 2 hay cả 3 tham số @MãPB, @TênPB, và @ChiNhánh là NULL
- @ChiNhánh chứa giá trị khác “Sài gòn” và “Hà nội”
- @MãPB đã có (hay bị trùng MãPB)

Sau đó dùng 2 stored procedure này để TEST thêm các hàng dữ liệu là:

Giá trị 3 tham số @MãPB, @TênPB, và @ChiNhánh	Trường hợp thử	Kết quả cần có
(NULL,N'Bảo hành',N'Sài gòn')	Không thêm được PB vì mã PB là NULL	→ Không thêm dữ liệu và hiện báo lỗi “Không thêm dữ liệu được vì không có giá trị mã PB!”
(N'PB07', N'Bảo hành',N'Sài gòn')	Thêm được PB	→ Thêm dữ liệu và hiện thông báo “Thêm dữ liệu thành công!”
(N'PB08',N'Tư vấn',N'Hà nội')	Thêm được PB	→ Thêm dữ liệu và hiện thông báo “Thêm dữ liệu thành công!”
(N'PB01',N'Kho vận',N'Hà nội')	Không thêm được PB vì bị trùng mã NXB	→ Không thêm dữ liệu và hiện báo lỗi “Không thêm dữ liệu được vì trùng mã phòng ban!”
(N'PB09',N'Nghiên cứu',N'Cần thơ')	Không thêm được PB vì chi nhánh không hợp lệ	→ Không thêm dữ liệu và hiện báo lỗi “Không thêm được PB vì chi nhánh không hợp lệ!”

**Câu 4:** Viết code SQL để tạo 2 stored procedure tên **SuaPhongBanMuc1** và **SuaPhongBanMuc2** để sửa (update) 2 cột TênPB và ChiNhánh (không sửa cột MãPB). Các tham số vào là @MãPB, @TênPB, và @ChiNhánh, trong đó @MãPB để xác định hàng dữ liệu cần sửa.

Yêu cầu: phải có báo chi tiết việc đã thực hiện khi sửa dữ liệu thành công, hay báo lỗi chi tiết (print ra màn hình) và không sửa dữ liệu khi gặp các trường hợp ngoại lệ sau:

- Có @MãPB là NULL
- Không tìm thấy @MãPB để sửa dữ liệu
- Có @ChiNhánh là NULL hay chứa giá trị khác “Sài gòn” và “Hà nội”

**Chú ý:** nếu sửa mức 2 mà có dời dữ liệu phòng ban thì cũng phải dời nhân viên của phòng ban đó sang phân mảnh khác tương ứng. Khi dời dữ liệu thì phải thông báo dời như thế nào trong thông báo sửa dữ liệu thành công.

Sau đó dùng 2 stored procedure này để TEST sửa các hàng dữ liệu:

Giá trị 3 tham số @MãPB, @TênPB, và @ChiNhánh	Trường hợp thử	Kết quả cần có
(NULL, N'Thiết kế', N'Sài gòn')	trường hợp có @MãPB là NULL	→ Không sửa dữ liệu và hiện báo lỗi “Không sửa dữ liệu được vì không có giá trị mã phòng ban!”
(N'PB01', N'Nghiên cứu', NULL)	trường hợp có @ChiNhánh là NULL	→ Không sửa dữ liệu và hiện báo lỗi “Không sửa dữ liệu được vì không có giá trị chi nhánh!”
(N'PB01', N'Nghiên cứu', N'Cần thơ')	trường hợp @ChiNhánh chứa giá trị khác “Sài gòn” và “Hà nội”	→ Không sửa dữ liệu và hiện báo lỗi “Không sửa dữ liệu được vì giá trị chi nhánh không hợp lệ!”
(N'PB01', N'Nghiên cứu', N'Sài gòn')	trường hợp không đổi chi nhánh	→ Sửa được dữ liệu và hiện thông báo thành công và nêu công việc đã thực hiện.
(N'PB02', N'Phát triển', N'Hà nội')	trường hợp có đổi loại hình	→ Sửa được dữ liệu và hiện thông báo thành công và nêu công việc đã thực hiện.
(N'PB06', N'Tài chính', N'Sài gòn')	trường hợp có đổi loại hình	→ Sửa được dữ liệu và hiện thông báo thành công và nêu công việc đã thực hiện.
(N'PB09', N'Kỹ thuật', N'Sài gòn')	trường hợp không tìm thấy mã phòng ban	→ Không sửa dữ liệu và hiện báo lỗi “Không sửa dữ liệu được vì không tìm thấy có giá trị mã phòng ban!”

**HẾT**

```

--HƯỚNG DẪN LÀM BÀI TẬP 2--
--Câu 1:
--Tạo phân mảnh ngang chính Phòng ban
use QLNhanVien
go
create proc TaoPhanManhPB
as
begin
select * into PB1
from [dbo].[PhongBan]
where ChiNhanh = N'Sài Gòn'

select * into PB2
from [dbo].[PhongBan]
where ChiNhanh <> N'Sài Gòn'
end

exec TaoPhanManhPB

--Tạo phân mảnh ngang dẫn xuất Nhân Viên
use QLNhanVien
go
create proc TaoPhanManhNV
as
begin
select nv.* into NV1
from [dbo].[NhanVien] nv, [dbo].[PB1] pb
where nv.MaPB = pb.MaPB

select nv.* into NV2
from [dbo].[NhanVien] nv, [dbo].[PB2] pb
where nv.MaPB = pb.MaPB
end

exec TaoPhanManhNV

--Câu 2
--Tạo proc DSNhanVienMuc1
use QLNhanVien
go
create proc DSNhanVienMuc1 @TenPB nvarchar(10)
as
begin
if @TenPB is not null
if (@TenPB in (select TenPB
from [dbo].[PhongBan]
where TenPB = @TenPB))
select nv.*, pb.TenPB, pb.ChiNhanh
from [dbo].[PhongBan] pb, [dbo].[NhanVien] nv
where pb.MaPB = nv.MaPB and pb.TenPB = @TenPB
else
print('Không tìm thấy tên phòng ban')
else
print('Không nhập tên phòng ban')
end
--Kiểm tra nhập đúng tên phòng ban
exec DSNhanVienMuc1 N'Thiết Kế'
exec DSNhanVienMuc1 N'Kế toán'

```

```

--Kiểm tra tên phòng ban không có trong CSDL
exec DSNhanVienMuc1 N'Kỹ thuật'
--Kiểm tra tên phòng ban nhập vào là NULL
exec DSNhanVienMuc1 NULL
go
-- Tạo proc DSNhanVienMuc2
create proc DSNhanVienMuc2 @TenPB nvarchar(10)
as
begin
if @TenPB is not null
if(@TenPB in (select TenPB from [dbo].[PB1] where TenPB = @TenPB))
select nv.*, pb.TenPB, pb.ChiNhanh
from [dbo].[PB1] pb, [dbo].[NhanVien] nv
where pb.MaPB = nv.MaPB and pb.TenPB = @TenPB
else if(@TenPB in (select TenPB from [dbo].[PB2] where TenPB = @TenPB))
select nv.*, pb.TenPB, pb.ChiNhanh
from [dbo].[PB2] pb, [dbo].[NhanVien] nv
where pb.MaPB = nv.MaPB and pb.TenPB = @TenPB
else
print('Không tìm thấy tên phòng ban')
else
print('Không nhập tên phòng ban')
end

--Kiểm tra nhập đúng tên phòng ban
exec DSNhanVienMuc2 N'Thiết Kế'
exec DSNhanVienMuc2 N'Kế toán'
--Kiểm tra tên phòng ban không có trong CSDL
exec DSNhanVienMuc2 N'Kỹ thuật'
--Kiểm tra tên phòng ban nhập vào là NULL
exec DSNhanVienMuc2 NULL
go

-- Câu 3:
-- Tạo pro ThemPhongBanMuc1
create proc ThemPhongBanMuc1 @MaPB nvarchar(10), @TenPB nvarchar(50), @ChiNhanh
nvarchar(50)
as
begin
if(@MaPB is not null and @TenPB is not null and @ChiNhanh is not null)
if(@MaPB not in (select MaPB from [dbo].[PhongBan] where MaPB = @MaPB))
if(@ChiNhanh in (select ChiNhanh from [dbo].[PhongBan] where ChiNhanh = ChiNhanh))
INSERT INTO PhongBan (MaPB, TenPB, ChiNhanh)
VALUES (@MaPB,@TenPB,@ChiNhanh);
else
print('Không thêm được PB vì chi nhánh không hợp lệ')
else
print('Không thêm dữ liệu được vì trùng mã phòng ban!')
else
print('Không thêm giá trị được vì có cột không có dữ liệu')
end
--Kiểm thử
exec ThemPhongBanMuc1 NULL,N'Bảo hành',N'Sài gòn'
exec ThemPhongBanMuc1 N'PB07', N'Bảo hành',N'Sài gòn'
exec ThemPhongBanMuc1 N'PB08',N'Tư vấn',N'Hà nội'
exec ThemPhongBanMuc1 N'PB01',N'Kho vận',N'Hà nội'
exec ThemPhongBanMuc1 N'PB09',N'Nghiên cứu',N'Cần thơ'

```

```

--Tạo proc ThemPhongBanMuc2
create proc ThemPhongBanMuc2 @MaPB nvarchar(10), @TenPB nvarchar(50), @ChiNhanh
nvarchar(50)
as
begin
if(@MaPB is not null and @TenPB is not null and @ChiNhanh is not null)
if(@MaPB not in (select MaPB from [dbo].[PB1] where MaPB = @MaPB) and @MaPB not in
(select MaPB from [dbo].[PB2] where MaPB = @MaPB))
if(@ChiNhanh in (select ChiNhanh from [dbo].[PB1] where ChiNhanh = ChiNhanh))
INSERT INTO [dbo].[PB1] (MaPB, TenPB, ChiNhanh)
VALUES (@MaPB,@TenPB,@ChiNhanh);
else if ( @ChiNhanh in (select ChiNhanh from [dbo].[PB2] where ChiNhanh = ChiNhanh))
INSERT INTO [dbo].[PB2] (MaPB, TenPB, ChiNhanh)
VALUES (@MaPB,@TenPB,@ChiNhanh);
else
print('Không thêm được PB vì chi nhánh không hợp lệ')
else
print('Không thêm dữ liệu được vì trùng mã phòng ban!')
else
print('Không thêm giá trị được vì có cột không có dữ liệu')
end
--Kiểm thử
exec ThemPhongBanMuc2 NULL,N'Bảo hành',N'Sài gòn'
exec ThemPhongBanMuc2 N'PB07', N'Bảo hành',N'Sài gòn'
exec ThemPhongBanMuc2 N'PB08',N'Tư vấn',N'Hà nội'
exec ThemPhongBanMuc2 N'PB01',N'Kho vận',N'Hà nội'
exec ThemPhongBanMuc2 N'PB09',N'Nghiên cứu',N'Cần thơ'

--Câu 4:
-- tạo proc SuaPhongBanMuc1
create proc SuaPhongBanMuc1 @MaPB nvarchar(10), @TenPB nvarchar(50), @ChiNhanh
nvarchar(50)
as
begin
if (@MaPB is not null)
if(@ChiNhanh is not null)
if(@ChiNhanh in (select ChiNhanh from [dbo].[PhongBan] where ChiNhanh = ChiNhanh))
if(@MaPB in (select MaPB from [dbo].[PhongBan] where MaPB = @MaPB))
UPDATE [dbo].[PhongBan]
SET TenPB = @TenPB, ChiNhanh = @ChiNhanh
WHERE MaPB = @MaPB;
else
print('Không sửa dữ liệu được vì không tìm thấy có giá trị mã phòng ban!')
else
print('Không sửa dữ liệu được vì giá trị chi nhánh không hợp lệ!')
else
print('Không sửa dữ liệu được vì không có giá trị chi nhánh!')
else
print('Không sửa dữ liệu được vì không có giá trị mã phòng ban!')
end

--Kiểm thử
exec SuaPhongBanMuc1 NULL, N'Thiết kế',N'Sài gòn'
exec SuaPhongBanMuc1 N'PB01', N'Nghiên cứu', NULL
exec SuaPhongBanMuc1 N'PB01', N'Nghiên cứu', N'Cần thơ'
exec SuaPhongBanMuc1 N'PB01',N'Nghiên cứu', N'Sài gòn'
exec SuaPhongBanMuc1 N'PB02',N'Phát triển',N'Hà nội'
exec SuaPhongBanMuc1 N'PB06',N'Tài chính',N'Sài gòn'

```

`exec SuaPhongBanMuc1 N'PB09',N'Kỹ thuật',N'Sài gòn'`