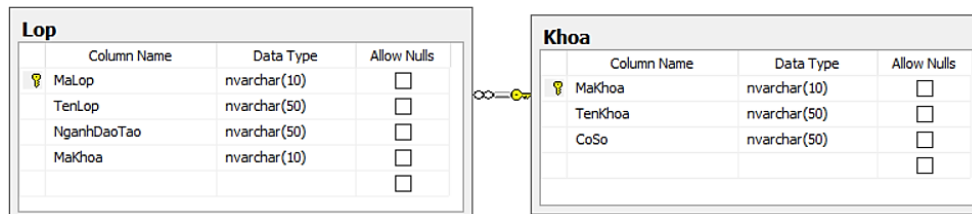


ÔN TẬP GIỮA KỲ 4

1. SV tái tạo (restore) CSDL **QLTruongDH** từ file “**QLTruongDH_DeThi.bak**” (đề cho) để làm bài.
2. SV thực hiện tất cả các câu đều phải có đủ 2 phần: **tạo** stored procedure và **test** stored procedure.
3. Sinh viên phải tự tạo đủ các trường hợp để kiểm thử, chứng minh là kết quả lập trình là đúng trong tất cả các trường hợp đề cho.

Cho CSDL **Quản lý trường đại học (QLTruongDH)** có 2 bảng như sau:



Lop(MaLop, TenLop, NganhDaoTao, #MaKhoa)

Khoa(MaKhoa, TenKhoa, CoSo)

Khóa chính: là MaKhoa và MaLop đã được gạch dưới

Khóa ngoại: Lop.MaKhoa là khóa ngoại tham khảo đến Khoa.MaKhoa

Câu 1: Hãy viết Stored procedure tên: **TaoPM_Doc_Khoa** để tạo và lấy dữ liệu cho **2 phân mảnh dọc** từ bảng **Khoa**, biết thiết kế của hai phân mảnh dọc là:

Khoa_Doc1(MaKhoa, TenKhoa)

Khoa_Doc2(MaKhoa, CoSo)

Câu 2: Hãy viết Stored procedure tên: **XemKhoa_Doc** có một tham số vào là @CoSo. Procedure này sẽ lập danh sách 3 cột là **MaKhoa, TenKhoa, CoSo** từ 2 phân mảnh dọc của bảng Khoa.

Yêu cầu:

- Nếu @CoSo là NULL: lập danh sách tất cả các khoa
- Có @CoSo không hợp lệ: báo lỗi
- Không tìm thấy giá trị @CoSo: báo lỗi

Sinh viên phải tự tạo đủ các trường hợp để kiểm thử, chứng minh kết quả lập trình là đúng trong tất cả các trường hợp.

Câu 3: Tạo stored procedure tên **ThemKhoa_Doc** để thêm dữ liệu trong **hai phân mảnh dọc của bảng Khoa**. Các tham số vào là @MaKhoa, @TenKhoa, và @CoSo

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

- Có @MaKhoa là NULL, hay @TenKhoa là NULL, hay @CoSo là NULL
- Có @CoSo chứa giá trị khác “Sài Gòn” và “Bình Dương”
- Bị trùng mã khoa nên không thêm dữ liệu được

Sinh viên phải tự tạo đủ các trường hợp để kiểm thử, chứng minh kết quả lập trình là đúng trong tất cả các trường hợp.

Câu 4: Hãy viết 2 Stored procedure tên: **TaoPM_Ngang_Khoa** để tạo và lấy dữ liệu cho các phân mảnh của bảng **Khoa**; và Stored procedure **TaoPM_Ngang_Lop** để tạo và lấy dữ liệu cho các phân mảnh của bảng **Lop**, biết:

- Bảng **Khoa** được **phân mảnh ngang chính** dựa vào cột **CoSo (cơ sở)**. Biết cột **CoSo** luôn có dữ liệu (NOT NULL) và chỉ có một trong hai giá trị là “Sài Gòn” và “Bình Dương”.
- Bảng **Lop** được **phân mảnh ngang dẫn xuất** theo các phân mảnh của bảng **Khoa**.

Câu 5: Tạo stored procedure tên **XemKhoa_Ngang** có một tham số vào là @CoSo. Procedure này sẽ lập danh sách 3 cột là **MaKhoa, TenKhoa, CoSo** từ 2 phân mảnh ngang chính của bảng **Khoa**.

Yêu cầu:

- Nếu @CoSo là NULL: lập danh sách tất cả các khoa
- Có @CoSo không hợp lệ: báo lỗi
- Không tìm thấy giá trị @CoSo: báo lỗi

Sinh viên phải tự tạo đủ các trường hợp để kiểm thử, chứng minh kết quả lập trình là đúng trong tất cả các trường hợp.

Câu 6: Tạo stored procedure tên **SuaKhoa_Ngang** để sửa dữ liệu (update) của các **phân mảnh ngang** của bảng **Khoa** tại 2 cột TenKhoa và CoSo (không sửa cột MaKhoa). Các tham số vào là @MaKhoa, @TenKhoa, và @CoSo, trong đó @MaKhoa để xác định hàng dữ liệu cần sửa.

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

- Có @MaKhoa là NULL, hay @TenKhoa là NULL, hay @CoSo là NULL
- Không tìm thấy @MaKhoa để sửa dữ liệu
- @CoSo chứa giá trị khác “Sài Gòn” và “Bình Dương”

Chú ý: nếu sửa khoa mà có dời dữ liệu khoa sang phân mảnh khác thì cũng phải dời lớp của khoa 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. **Sinh viên phải tự tạo đủ các trường hợp để kiểm thử**, chứng minh kết quả lập trình là đúng trong tất cả các trường hợp.

HƯỚNG DẪN LÀM BÀI

```
USE [master]
RESTORE DATABASE [QLTruongDH] FROM DISK = N'D:\QLTruongDH_DeThi.bak' WITH FILE = 1,
MOVE N'QL_TruongDH' TO N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\QL_TruongDH.mdf', MOVE N'QL_TruongDH_log' TO
N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\QL_TruongDH_log.ldf', NOUNLOAD, STATS = 5

GO

use QLTruongDH
go

--cau 1
create proc TaoPM_Doc_Khoa
as
begin
select MaKhoa, TenKhoa into Khoa_Doc1 from Khoa
select MaKhoa, CoSo into Khoa_Doc2 from Khoa
end
go

exec TaoPM_Doc_Khoa
go

--cau2

create proc XemKhoa_Doc(@CoSo nvarchar(50))
as
begin
begin try
if(@CoSo is null)
begin
select a.MaKhoa, TenKhoa, CoSo from Khoa_Doc1 a join Khoa_Doc2 b on a.MaKhoa = b.MaKhoa
return
end
if(not exists(select CoSo from Khoa_Doc2 where CoSo = @CoSo))
raiserror(N'Co So khong hop le', 16,1)
select a.MaKhoa, TenKhoa, CoSo from Khoa_Doc1 a join Khoa_Doc2 b on a.MaKhoa = b.MaKhoa
where CoSo = @CoSo
end try
begin catch
DECLARE @ErrorMessage NVARCHAR(4000);
DECLARE @ErrorSeverity INT;
DECLARE @ErrorState INT;

SELECT
    @ErrorMessage = ERROR_MESSAGE(),
```

```

        @ErrorSeverity = ERROR_SEVERITY(),
        @ErrorState = ERROR_STATE();

-- Use RAISERROR inside the CATCH block to return error
-- information about the original error that caused
-- execution to jump to the CATCH block.
RAISERROR (@ErrorMessage, -- Message text.
           @ErrorSeverity, -- Severity.
           @ErrorState -- State.
           );
end catch
end
go

exec XemKhoa_Doc null
go
exec XemKhoa_Doc N'@@@#'
go
exec XemKhoa_Doc N'Bình Dương'
go
exec XemKhoa_Doc N'Sài Gòn'
go

drop proc XemKhoa_Doc
go

--cau3
create proc ThemKhoa_Doc (@MaKhoa nvarchar(10), @TenKhoa nvarchar(50), @CoSo
nvarchar(50))
as
begin
begin try
if(@MaKhoa is null and @TenKhoa is null and @CoSo is null)
raiserror(N'Mã Khoa null, Tên Khoa null, Cơ Sở null', 16,1)
if(@MaKhoa is null)
raiserror(N'Mã Khoa null', 16,1)
if(@TenKhoa is null)
raiserror(N'Tên Khoa null', 16,1)
if(@CoSo is null)
raiserror(N'Cơ Sở null', 16,1)
if(@CoSo != N'Bình Dương' and @CoSo != N'Sài Gòn')
raiserror(N'Cơ Sở khác Bình Dương và Sài Gòn', 16,1)
if(exists(select MaKhoa from Khoa_Doc1 where MaKhoa = @MaKhoa))
raiserror(N'Trùng Mã Khoa không thêm được', 16,1)
insert into Khoa_Doc1 values(@MaKhoa, @TenKhoa)
insert into Khoa_Doc2 values(@MaKhoa, @CoSo)
print N'Thêm thành công'
end try
begin catch
declare @t nvarchar(4000)
declare @s int
declare @st int
select
@t = ERROR_MESSAGE(),
@s = ERROR_SEVERITY(),
@st = ERROR_STATE()
raiserror(@t,@s,@st)
end catch
end
end

```

```

go

exec ThemKhoa_Doc null, 'saa', 'saas'
go
exec ThemKhoa_Doc 'saa', null, 'saas'
go
exec ThemKhoa_Doc 'saa', 'saas', null
go
exec ThemKhoa_Doc null, null, null
go
exec ThemKhoa_Doc N'NNH', N'Ngoại ngữ học', N'Cần Thơ'
go
exec ThemKhoa_Doc N'QTKD', N'Ngoại ngữ học', N'Sài Gòn'
go
exec ThemKhoa_Doc N'NNH', N'Ngoại ngữ học', N'Sài Gòn'
go

--cau 4
create proc TaoPM_Ngang_Khoa
as
begin
select * into Khoa_Ngang1 from Khoa where CoSo = N'Bình Dương'
select * into Khoa_Ngang2 from Khoa where CoSo = N'Sài Gòn'
end
go

exec TaoPM_Ngang_Khoa
go

create proc TaoPM_Ngang_Lop
as
begin
select * into Lop_Ngang1 from Lop where MaKhoa in (select MaKhoa from Khoa_Ngang1)
select * into Lop_Ngang2 from Lop where MaKhoa in (select MaKhoa from Khoa_Ngang2)
end
go

exec TaoPM_Ngang_Lop
go

--cau5
create proc XemKhoa_Ngang(@CoSo nvarchar(50))
as
begin
begin try
if(@CoSo is null)
raiserror(N'Cơ sở null', 16,1)
if(not exists(select CoSo from Khoa_Ngang1 where CoSo = @CoSo) and not exists(select CoSo
from Khoa_Ngang2 where CoSo = @CoSo))
raiserror(N'Không tìm thấy cơ sở', 16,1)
if(@CoSo = N'Bình Dương')
select * from Khoa_Ngang1
else
select * from Khoa_Ngang2
end try
begin catch
declare @t nvarchar(4000)
declare @s int
declare @st int

```

```

select
@t = ERROR_MESSAGE(),
@s = ERROR_SEVERITY(),
@st = ERROR_STATE()
raiserror(@t,@s,@st)
end catch
end
go

exec XemKhoa_Ngang null
go
exec XemKhoa_Ngang N'Cần Thơ'
go
exec XemKhoa_Ngang N'Bình Dương'
go
exec XemKhoa_Ngang N'Sài Gòn'
go

--cau6
create proc SuaKhoa_Ngang(@MaKhoa nvarchar(10), @TenKhoa nvarchar(50), @CoSo
nvarchar(50))
as
begin
begin try
if(@MaKhoa is null and @TenKhoa is null and @CoSo is null)
raiserror(N'Mã Khoa null, Tên Khoa null, Cơ Sở null', 16,1)
if(@MaKhoa is null)
raiserror(N'Mã Khoa null', 16,1)
if(@TenKhoa is null)
raiserror(N'Tên Khoa null', 16,1)
if(@CoSo is null)
raiserror(N'Cơ Sở null', 16,1)
if(not exists(select MaKhoa from Khoa_Ngang1 where MaKhoa = @MaKhoa) and not
exists(select MaKhoa from Khoa_Ngang2 where MaKhoa = @MaKhoa))
raiserror(N'Không tìm thấy Mã Khoa', 16,1)
if(@CoSo != N'Bình Dương' and @CoSo != N'Sài Gòn')
raiserror(N'Cơ Sở khác Bình Dương và Sài Gòn', 16,1)
if(exists(select MaKhoa from Khoa_Ngang1 where MaKhoa = @MaKhoa))
begin
if(@CoSo = N'Bình Dương')
begin
update Khoa_Ngang1 set TenKhoa = @TenKhoa where MaKhoa = @MaKhoa
print N'đã sửa tên khoa thành công'
return
end
else
begin
update Khoa_Ngang1 set TenKhoa = @TenKhoa, CoSo = @CoSo where MaKhoa = @MaKhoa
print N'Đã sửa tên khoa và cơ sở thành công'
insert into Khoa_Ngang2 select * from Khoa_Ngang1 where MaKhoa = @MaKhoa
print N'Đã thêm khoa muốn sửa vào khoa 2'
insert into Lop_Ngang2 select * from Lop_Ngang1 where MaKhoa = @MaKhoa
print N'Đã thêm các lớp của khoa đó vào lớp 2'
delete from Khoa_Ngang1 where MaKhoa = @MaKhoa
print N'Đã xóa khoa được sửa ở khoa 1'
delete from Lop_Ngang1 where MaKhoa = @MaKhoa
print N'Đã xóa các lớp của khoa ở lớp 1'
return
end
end
end

```

```

end
if(exists(select MaKhoa from Khoa_Ngang2 where MaKhoa = @MaKhoa))
begin
if(@CoSo = N'Sài Gòn')
begin
update Khoa_Ngang2 set TenKhoa = @TenKhoa where MaKhoa = @MaKhoa
print N'Đã sửa tên khoa thành công'
return
end
else
begin
update Khoa_Ngang2 set TenKhoa = @TenKhoa, CoSo = @CoSo where MaKhoa = @MaKhoa
print N'Đã sửa tên khoa và cơ sở thành công'
insert into Khoa_Ngang1 select * from Khoa_Ngang2 where MaKhoa = @MaKhoa
print N'Đã thêm khoa muốn sửa vào khoa 1'
insert into Lop_Ngang1 select * from Lop_Ngang2 where MaKhoa = @MaKhoa
print N'Đã thêm các lớp của khoa đó vào lớp 1'
delete from Khoa_Ngang2 where MaKhoa = @MaKhoa
print N'Đã xóa khoa được sửa ở khoa 2'
delete from Lop_Ngang2 where MaKhoa = @MaKhoa
print N'Đã xóa các lớp của khoa ở lớp 2'
return
end
end
end try
begin catch
declare @t nvarchar(4000)
declare @s int
declare @st int
select
@t = ERROR_MESSAGE(),
@s = ERROR_SEVERITY(),
@st = ERROR_STATE()
raiserror(@t,@s,@st)
end catch
end
go

exec SuaKhoa_Ngang null, null, null
go
exec SuaKhoa_Ngang null, 'sad', 'asdas'
go
exec SuaKhoa_Ngang 'null', null, 'null'
go
exec SuaKhoa_Ngang 'null', 'null', null
go
exec SuaKhoa_Ngang N'NNH', N'Ngôn ngữ học', N'Cần Thơ'
go
exec SuaKhoa_Ngang N'CNSH', N'Ngôn ngữ học', N'Bình Dương'
go
exec SuaKhoa_Ngang N'CNSH', N'Ngôn ngữ học2', N'Sài Gòn'
go
exec SuaKhoa_Ngang N'CNSH', N'Ngôn ngữ học', N'Bình Dương'
go

```