

Review for midterm exam

Exercise 1:

1. Thêm 1 cột evaluation vào bảng Student: evaluation nvarchar(50)

2. Viết Store procedure trả về @evaluation nvarchar(50) của @StudentID như sau:

-Bước 1: kiểm tra sự tồn tại của @StudentID trong bảng Student bằng việc gọi store procedure Step1 nhận vào @StudentID trả về 1 hoặc 0 tương ứng tồn tại hay không.

-Bước 2: nếu số tín chỉ mà @StudentID đã tích lũy nhỏ hơn 20 thì đánh giá là "Không Đạt" ngược lại chuyển sang bước 3

-Bước 3: nếu điểm @StudentID trong môn Cơ sở dữ liệu hơn 8 điểm thì @evaluation là "giỏi" ngược lại @evaluation là "Khá"

-Bước 4: Cập nhật thuộc tính evaluation của Student và xuất ra kết quả.

3. Viết store cập nhật evaluation của tất cả Student

--Câu 1:

```
alter table student
add evaluation nvarchar(50)
```

--Câu 2:

```
create proc step1
    @studentID varchar(50), @result int output
as
begin
```

```
    declare @i int
    select @i=count(*) from student
    where ID= @studentID
    if(@i=0)
        set @result =0
    else
        set @result=1
```

```
end
```

```
go
```

```
create proc FillEvaluation
```

```

    @studentID          varchar(50),          @evaluation
nvarchar(50) output
as
begin
    declare @result int
    exec step1 @studentID,@result output
    if(@result = 0)
    begin
        print 'ko ton tai student ID'
        return
    end
    declare @creditsum int
    select @creditsum=sum(sub.credits)
    from result r, subject sub
    where          r.subjectID=sub.ID          and
r.studentID=@studentID and r.mark>=5 and
r.times>=all(select r1.times from result r1
where          r1.studentID=r.studentID          and
r1.subjectID=r.subjectID)
    if(@creditsum <20)
        set @evaluation=N'Không Đạt'
    else
    begin
        declare @mark float
        select @mark=r.mark
        from result r, subject sub
        where          r.subjectID=sub.ID          and
r.studentID=@studentID          and sub.name= N'Cơ
sở dữ liệu'          and          r.times>=all(select
r1.times          from          result          r1          where
r1.studentID=r.studentID          and
r1.subjectID=r.subjectID)
        if(@mark >=8)

```

```

        set @evaluation=N'Giỏi'
    else
        set @evaluation=N'Khá'
    end
    update student set evaluation =
    @evaluation where ID=@studentID
    print @evaluation
end
declare @evaluation nvarchar(50)
exec FillEvaluation 'HV000001',@evaluation
output
--Câu 3:
create proc FillToAllStu
as
begin
    declare c cursor for select ID from student
    open c
    declare @ID varchar(20)
    fetch next from c into @ID
    while (@@fetch_status=0)
    begin
        declare @evaluation nvarchar(50)
        exec FillEvaluation @ID,@evaluation
    output
        fetch next from c into @ID
    end
    close c
    deallocate c
end
exec FillToAllStu
select * from student

```

-Exercise 2:

1. Thêm 1 cột evaluation vào bảng Teacher:
evaluation nvarchar(50)

2. Viết Store procedure trả về @evaluation
nvarchar(50) của @TeacherID như sau: -

Bước 1: kiểm tra sự tồn tại của @TeacherID trong
bảng Teacher bằng việc gọi store procedure
Step1 nhận vào @TeacherID trả về 1 hoặc 0 tương
ứng tồn tại hay không.

-Bước 2: Tổng số môn mà @TeacherID có **khả năng
dạy** nhỏ hơn 5 thì đánh giá là "Không Đạt" ngược
lại qua bước 3

-Bước 3: @TeacherID đã từng được phân công chủ
nhiệm thì @evaluation là "giỏi" ngược lại là
"khá"

-Bước 4: Cập nhật thuộc tính evaluation của
Teacher và xuất ra kết quả.

3. Viết store cập nhật evaluation của tất cả
Teacher.

```
Alter table Teacher
add evaluation nvarchar(50)
create proc step_1
    @teacherID varchar(50), @result int output
as
begin
    if (exists (select * from teacher where
ID=@teacherID))
        set @result=1
    else
        set @result=0
```

```

end

create proc FillTeacher
    @teacherID          varchar(50), @evaluation
nvarchar(50) output
as
begin
    declare @result int
    exec step_1 @teacherID, @result output
    if (@result=0)
    begin
        print 'ko ton tai'
        return
    end
    declare @i int
    select  @i=count(*)    from  ability  where
teacherID=@teacherID
    if (@i<5)
        set @evaluation=N'Không Đạt'
    else
    begin
        if (exists (select * from class where
managerID=@teacherID))
            set @evaluation=N'Giỏi'
        else
            set @evaluation=N'Khá'
        end
        update teacher set evaluation=@evaluation
where ID=@teacherID
        print @evaluation
    end

create proc FillAllTeacher

```

```

as
begin
    declare c cursor for select ID from Teacher
    open c
    declare @ID varchar(20)
    fetch next from c into @ID
    while (@@fetch_status=0)
    begin
        declare @evaluation nvarchar(50)
        exec FillTeacher @ID,@evaluation output
        fetch next from c into @ID
    end
    close c
    deallocate c
end
exec FillAllTeacher
select * from teacher

```

-Exercise 3

1.Thêm 1 cột evaluation vào bảng Subject:
evaluation nvarchar(50)

2.Viết Store procedure nhận @SubjectID làm tham số và xử lý như sau:

-Bước 1: Tính tổng số sv đã thi đậu @SubjectID

-Bước 2: Tính tổng số sv đã thi @SubjectID

-Bước 3: Nếu giá trị bước 1 $\geq 50\%$ giá trị bước 2 thì @evaluation là 'Đạt' ngược lại là 'Không Đạt'

-Bước 4: Cập nhật cột evaluation của SubjectID

```
Alter table Subject
add evaluation nvarchar(50)
create proc FillSubject
    @SubjectID varchar(50)
as
begin
    declare @i int
    select @i=count(*)
    from result r where r.subjectID=@SubjectID
    and r.mark>=5 and r.times>=all(select
r1.times from result r1 where
r1.studentID=r.studentID and
r1.subjectID=r.subjectID)

    declare @j int
    select @j=count(distinct r.studentID)
    from result r where r.subjectID=@SubjectID

    if (@i>@j/2.0)
```

```

        update Subject set evaluation=N'Không
Đạt' where ID=@SubjectID
    else
        update Subject set evaluation=N'Đạt'
where ID=@SubjectID
end
exec FillSubject 'MH00001'

```

-Exercise 4. Xuất ra bảng điểm của 1 @studentID như sau:

Họ Ten:

DTB:

Xếp Loại:

Kết quả học tập:

STT	Tên môn học	STC	Điểm
-----	-------------	-----	------

```

alter proc tinhdiemtrungbinh @studentID
varchar(20),@dtb float output
as begin

```

```

    select
@dtb=sum(kq.mark*mh.credits)/sum(mh.credits)
    from student hv,result kq, subject mh
    where hv.id=kq.studentID
    and kq.mark is not null
    and kq.subjectID=mh.ID
    and kq.times >=
    all( select kq1.times
    from result kq1 where
kq1.subjectID=kq.subjectID
    and kq1.studentID=kq.studentID)
    and hv.ID= @studentID

```



```

end
go

alter proc xuấtbangdiem @studentID
varchar(20)
as
begin
    declare @ten nvarchar(51)
    select @ten=name from student where
ID=@studentID
    print 'ho ten:' +@ten
    declare @dtb float
    exec tinhdiemtrungbinh @studentID,@dtb
output
    print 'DTB:' + cast(ROUND(@dtb,2) as
varchar(4))
    if(@dtb>=8)
        print 'Xep loai: Gioi'
    else
        if(@dtb>=7)
            print 'Xep loai: Khá'
        else
            print 'Xep loai: Trung Bình'
    print 'Ket qua hoc tap'
    DECLARE c CURSOR FOR
    SELECT m.name,m.credits,kq.mark
    FROM result kq, subject m
    WHERE kq.subjectID=m.ID
    and kq.studentID=@studentID
    and kq.mark is not null
    and kq.times >=
    all( select kq1.times

```

```

    from result kq1 where
kq1.subjectID=kq.subjectID
    and kq1.studentID=kq.studentID)
DECLARE @M NCHAR(40),@ST INT,@d float
OPEN c
FETCH NEXT FROM c INTO @M,@ST,@d
declare @STT int
Set @STT=1
WHILE @@FETCH_STATUS = 0
BEGIN
    print cast(@STT as char(2))+' ' +
    @M + ' ' + cast(@ST as char(2)) + ' ' +
    CAST(ROUND(@d,2) AS CHAR(4))
    FETCH NEXT FROM c INTO @M,@ST,@d
    set @STT=@STT+1
END
CLOSE c
DEALLOCATE c
end
go
exec xuatbangdiem 'HV000001 '

```

Exercise 5:

Viết stored procedure nhận vào thông tin một kết quả học tập (mã môn học, mã học viên, điểm) và đưa kết quả này vào CSDL theo quy trình sau:

- B1: Kiểm tra nếu mã học viên không tồn tại → thông báo lỗi học viên không tồn tại & kết thúc.
- B2: Kiểm tra nếu mã môn học không tồn tại → thông báo lỗi môn học không tồn tại & kết thúc.
- B3: Kiểm tra nếu điểm môn học không phải nằm trong đoạn [0, 10] → thông báo lỗi điểm không hợp lệ & kết thúc.
- B4: Phát sinh giá trị lần thi theo công thức: lần thi mới = lần thi gần nhất + 1.
- B5: Thêm kết quả vào CSDL & thông báo thành công.

```
create proc laylanthiganday
@mmh varchar(10), @mhv varchar(10), @lt int
output
as begin
    select @lt=isnull(max(kq1.times),0)
    from result kq1 where kq1.subjectID=@mmh
    and kq1.studentID=@mhv
end
go
create proc themkq
@mmh varchar(10), @mhv varchar(10), @d int
as begin
    if( not exists(select * from student where
ID=@mhv) )
    begin
        print('Loi ko ton tai hoc vien')
        return
    end
end
```

```

        if( not exists(select * from subject where
id=@mmh))
        begin
            print('Loi ko ton tai mon hoc')
            return
        end
        if( @d<0 or @d>10)
        begin
            print('diem ko hop le')
            return
        end
        declare @lt int
        exec laylanthiganday @mmh,@mhv,@lt output
        set @lt=@lt+1

        insert into result values
(@mhv,@mmh,@lt,@d)
        print 'them thanh cong'
    end
go

```