**Lab 9 : Crearea procedurilor stocate si a functiilor definite de utilizator**

**Task 1**

**Sa se creeze proceduri stocate in baza exercitiilor (2 exercitii) din capitolul 4. Parametrii de intrare trebuie sa corespunda criteriilor din clauzele WHERE ale exercitiilor respective.**

--Afisati numarul de studenti care au sustinut testul (Testul 2) la disciplina Baze de date in 2018

create procedure proc\_20

@Tip\_Evaluare varchar(50)

as

select count(Distinct s.Nume\_Student)

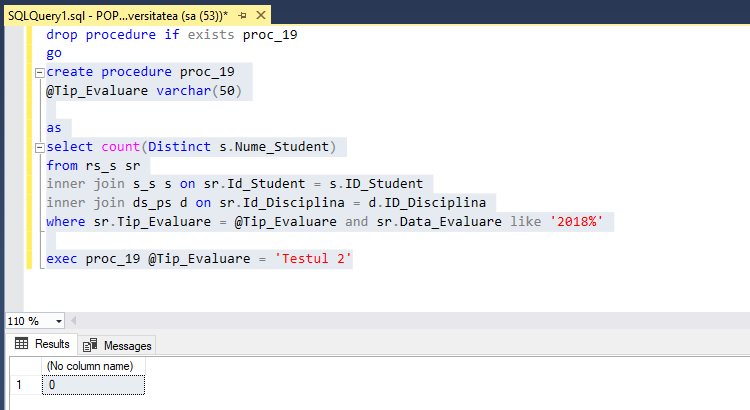
from rs\_s sr

inner join s\_s s on sr.Id\_Student = s.ID\_Student

inner join ds\_ps d on sr.Id\_Disciplina = d.ID\_Disciplina

where sr.Tip\_Evaluare = @Tip\_Evaluare and sr.Data\_Evaluare like '2018%'

exec proc\_20 @Tip\_Evaluare = 'Testul 2'



--39 Gasiti denumirile disciplinelor la care nu au sustinut examenul, in medie, peste 5% de studenti.

create procedure proc\_39

@Percentage float

as

select distinct d.Disciplina

from rs\_s sr

inner join ds\_ps d on sr.Id\_Disciplina = d.Id\_Disciplina

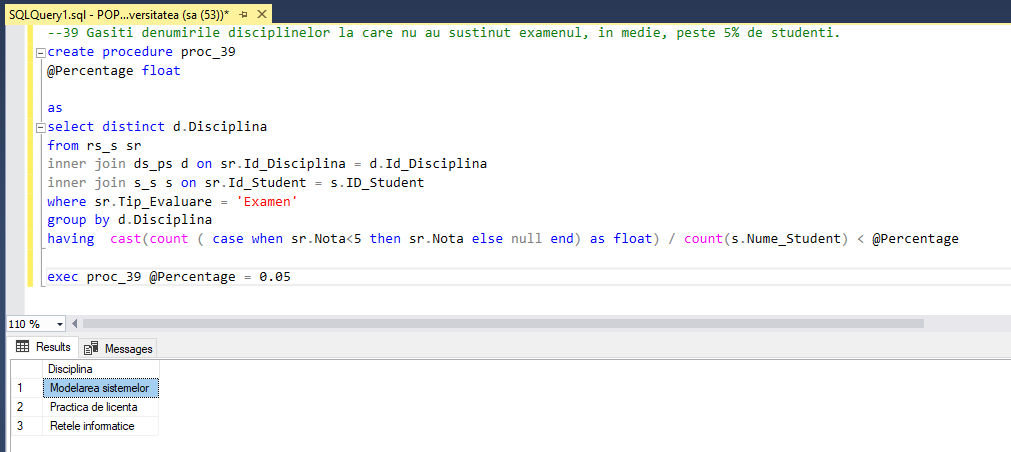
inner join s\_s s on sr.Id\_Student = s.ID\_Student

where sr.Tip\_Evaluare = 'Examen'

group by d.Disciplina

having cast(count ( case when sr.Nota<5 then sr.Nota else null end) as float) / count(s.Nume\_Student) < @Percentage

exec proc\_39 @Percentage = 0.05



**Task 2**

**Sa se creeze o procedura stocata, care nu are niciun parametru de intrare si poseda un parametru de iesire. Parametrul de iesire trebuie sa returneze numarul de studenti, care nu au sustinut cel putin o forma de evaluare (nota mai mica de 5 sau valoare NULL).**

create procedure proc2\_20

@Nr int = null output

as

select @Nr = count(Distinct s.Nume\_Student)

from rs\_s sr

inner join s\_s s on sr.Id\_Student = s.ID\_Student

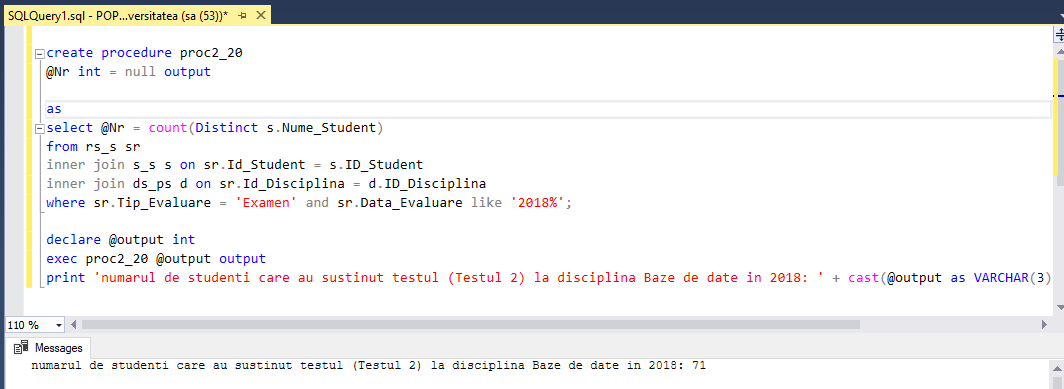
inner join ds\_ps d on sr.Id\_Disciplina = d.ID\_Disciplina

where sr.Tip\_Evaluare = 'Examen' and sr.Data\_Evaluare like '2018%';

declare @output int

exec proc2\_20 @output output

print 'numarul de studenti care au sustinut testul (Testul 2) la disciplina Baze de date in 2018: ' + cast(@output as VARCHAR(3))



**Task 3**

**Sa se creeze o procedura stocata, care ar insera in baza de date informatii despre un student nou. In calitate de parametri de intrare sa serveasca datele personale ale studentului nou si Cod\_Grupa. Sa se genereze toate intrarile-cheie necesare in tabelul studenti\_reusita. Notele de evaluare sa fie inserate ca NULL.**

create procedure addStudent

@nume varchar(60),

@prenume varchar(60),

@data date,

@adresa varchar(100),

@codGrupa char(10)

as

insert into s\_s

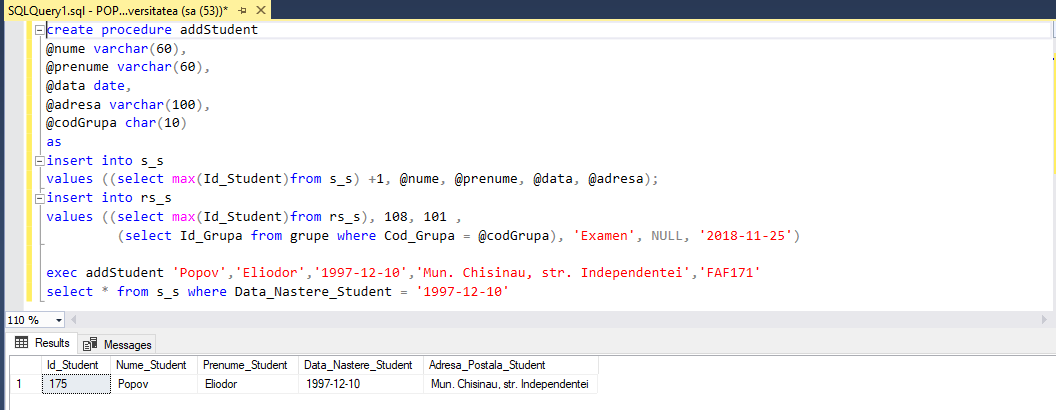
values ((select max(Id\_Student)from s\_s) +1, @nume, @prenume, @data, @adresa);

insert into rs\_s

values ((select max(Id\_Student)from rs\_s), 108, 101 ,

(select Id\_Grupa from grupe where Cod\_Grupa = @codGrupa), 'Examen', NULL, '2018-11-25')

exec addStudent 'Popov','Eliodor','1997-12-10','Mun. Chisinau, str. Independentei','FAF171'



**Task 4**

**Fie ca un profesor se elibereaza din functie la mijlocul semestrului. Sa se creeze o procedura stocata care ar reatribui inregistrarile din tabelul studenti\_reusita unui alt profesor. Parametri de intrare: numele si prenumele profesorului vechi, numele si prenumele profesorului nou, disciplina. in cazul in care datele inserate sunt incorecte sau incomplete, sa se afiseze un mesaj de avertizare.**

create procedure procedure4

@old\_last\_name VARCHAR(50),

@old\_first\_name VARCHAR(50),

@new\_last\_name VARCHAR(50),

@new\_first\_name VARCHAR(50),

@disciplina VARCHAR(50)

as

if(( select ds\_ps.Id\_Disciplina

from ds\_ps

where Disciplina = @disciplina) in (select distinct rs\_s.Id\_Disciplina

from rs\_s

where Id\_Profesor = (select pf\_cd.Id\_Profesor

from pf\_cd

where Nume\_Profesor = @old\_last\_name

and Prenume\_Profesor = @old\_first\_name)))

begin

update rs\_s

set Id\_Profesor = (select Id\_Profesor

from pf\_cd

where Nume\_Profesor = @new\_last\_name

and Prenume\_Profesor = @new\_first\_name)

where Id\_Profesor = (select Id\_profesor

from pf\_cd

where Nume\_Profesor = @old\_last\_name

and Prenume\_Profesor = @old\_first\_name)

end

else

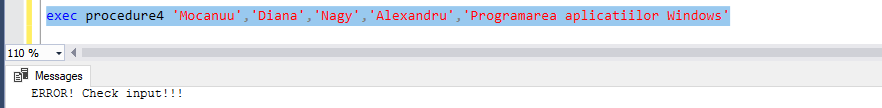
begin

print 'ERROR! Check input!!!'

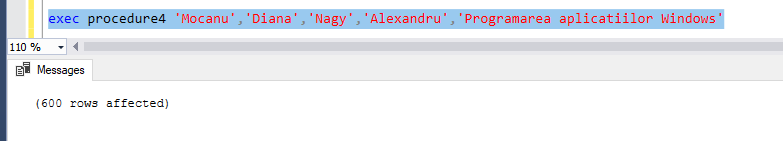
end

exec procedure4 'Mocanu','Diana','Nagy','Alexandru','Programarea aplicatiilor Windows'

**ERROR**



**SUCCESS**



**Task 5**

**Sa se creeze o procedura stocata care ar forma o lista cu primii 3 cei mai buni studenti la o disciplina, si acestor studenti sa le fie marita nota la examenul final cu un punct (nota maximala posibila este 10). In calitate de parametru de intrare, va servi denumirea disciplinei. Procedura sa returneze urmatoarele campuri: Cod\_Grupa, Nume\_Prenume\_Student, Disciplina, Nota\_ Veche, Nota\_Noua.**

create procedure procedure5

@disciplina VARCHAR(50)

as

declare @studenti\_lista table (Id\_Student int, Media float)

insert into @studenti\_lista

select top (3) rs\_s.Id\_Student, AVG(cast (Nota as float)) as Media

from rs\_s, ds\_ps

where ds\_ps.Id\_Disciplina = rs\_s.Id\_Disciplina

and Disciplina = @disciplina

group by rs\_s.Id\_Student

order by Media desc;

select cod\_grupa, s\_s.Id\_Student, CONCAT(nume\_student, ' ', Prenume\_Student) as Nume, Disciplina, nota AS Nota\_Veche, iif(nota > 9, 10, nota + 1) AS Nota\_Noua

from rs\_s, ds\_ps, grupe, s\_s

where ds\_ps.id\_disciplina = rs\_s.id\_disciplina

and grupe.Id\_Grupa = rs\_s.Id\_Grupa

and s\_s.Id\_Student = rs\_s.Id\_Student

and s\_s.Id\_Student in (select Id\_Student from @studenti\_lista)

and Disciplina = @disciplina

and Tip\_Evaluare = 'Examen';

declare @id\_discipl smallint = (select Id\_Disciplina

from ds\_ps

where Disciplina = @disciplina);

update rs\_s

set rs\_s.Nota = (CASE WHEN nota >= 9 THEN 10 ELSE nota + 1 END)

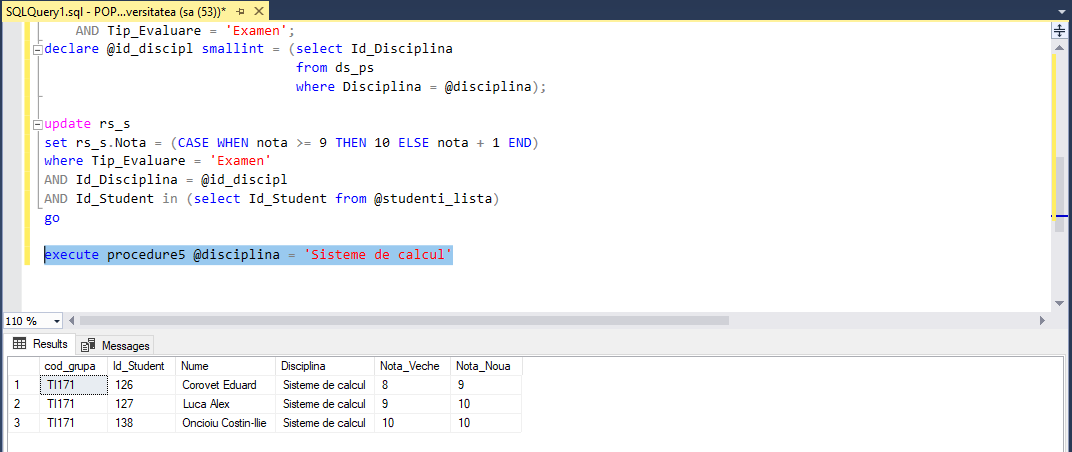
where Tip\_Evaluare = 'Examen'

and Id\_Disciplina = @id\_discipl

and Id\_Student in (select Id\_Student from @studenti\_lista)

go

execute procedure5 @disciplina = 'Sisteme de calcul'



**Task 6**

**Sa se creeze functii definite de utilizator in baza exercitiilor (2 exercitii) din capitolul 4. Parametrii de intrare trebuie sa corespunda criteriilor din clauzele WHERE ale exercitiilor respective.**

create function fun19 (@nume\_student VARCHAR(10), @reusita SMALLINT)

returns table

as

return

(

select distinct Nume\_Profesor,Prenume\_Profesor

from rs\_s sr

inner join pf\_cd p on sr.Id\_Profesor = p.Id\_Profesor

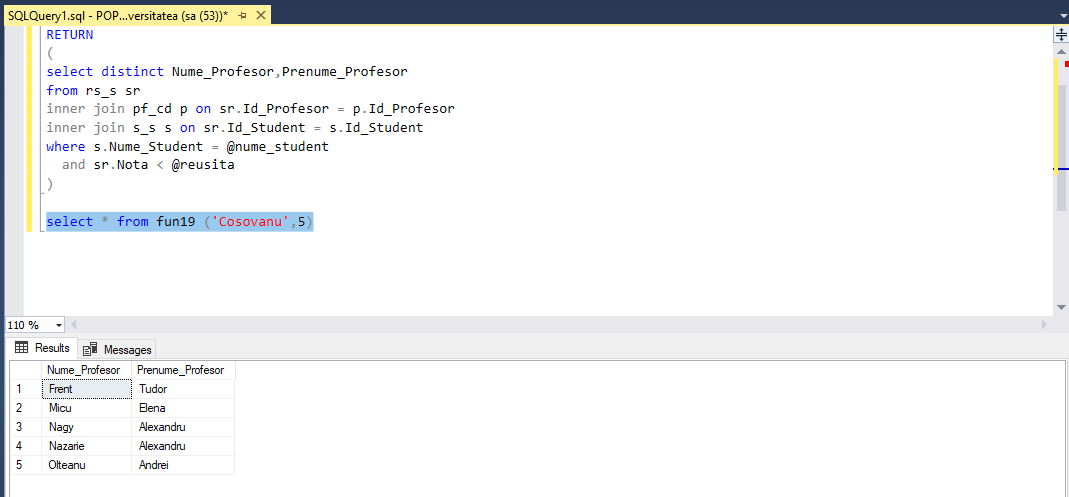
inner join s\_s s on sr.Id\_Student = s.Id\_Student

where s.Nume\_Student = @nume\_student

and sr.Nota < @reusita

)

select \* from fun19 ('Cosovanu',5)



create function fun39 (@Percentage float)

returns table

as

return

(

select distinct d.Disciplina

from rs\_s sr

inner join ds\_ps d on sr.Id\_Disciplina = d.Id\_Disciplina

inner join s\_s s on sr.Id\_Student = s.ID\_Student

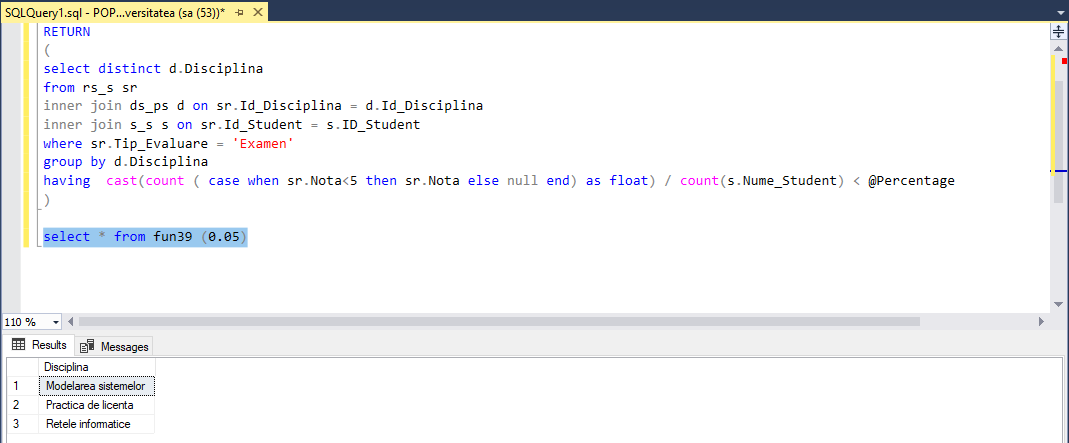
where sr.Tip\_Evaluare = 'Examen'

group by d.Disciplina

having cast(count ( case when sr.Nota<5 then sr.Nota else null end) as float) / count(s.Nume\_Student) < @Percentage

)

select \* from fun39 (0.05)



**Task 7**

**Sa se scrie functia care ar calcula varsta studentului. Sa se defineasca urmatorul format al functiei: <nume\_functie>(<Data\_Nastere\_Student>).**

create function fun7 (@data\_nasterii date )

returns int

begin

declare @varsta int

select @varsta = (select (year(getdate()) - year(@data\_nasterii) - case

when (month(@data\_nasterii) > month(getdate()))

or (month(@data\_nasterii) = month(getdate())

and day(@data\_nasterii) > day(getdate()))

THEN 1

ELSE 0

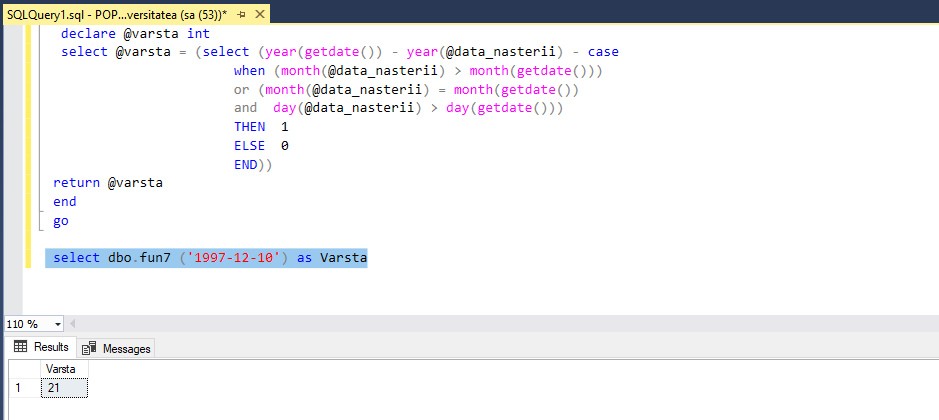
END))

return @varsta

end

go

select dbo.fun7 ('1997-12-10') as Varsta



**Task 8**

**Sa se creeze o functie definita de utilizator, care ar returna datele referitoare la reusita unui student. Se defineste urmatorul format al functiei : < nume\_functie > (<Nume\_Prenume\_Student>). Sa fie afisat tabelul cu urmatoarele campuri: Nume\_Prenume\_Student, Disticplina, Nota, Data\_Evaluare.**

create function fun8 (@st\_nume\_prenume VARCHAR(50))

returns table

as

return

(select concat(Nume\_Student, ' ',Prenume\_Student) as Student, Disciplina, Nota, Data\_Evaluare

from s\_s, ds\_ps, rs\_s

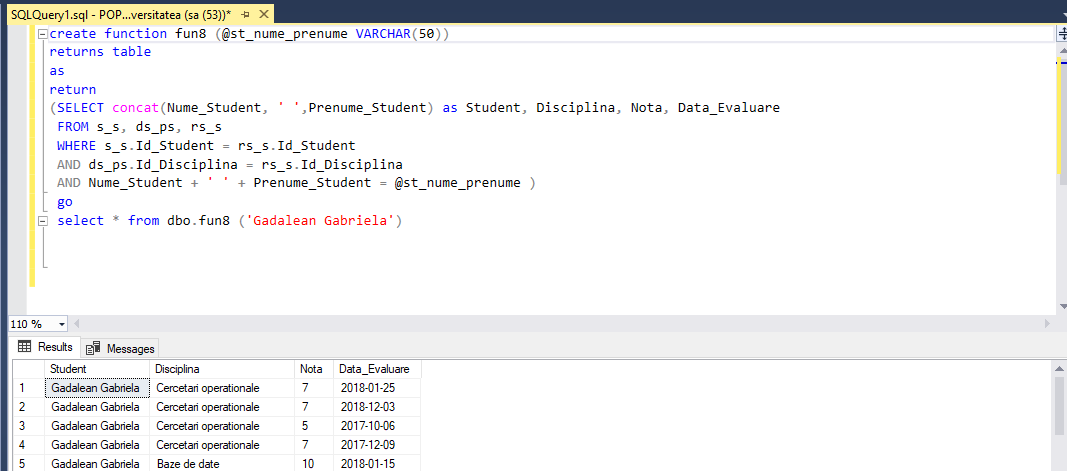
where s\_s.Id\_Student = rs\_s.Id\_Student

and ds\_ps.Id\_Disciplina = rs\_s.Id\_Disciplina

and Nume\_Student + ' ' + Prenume\_Student = @st\_nume\_prenume )

go

select \* from dbo.fun8 ('Gadalean Gabriela')



**Task 9**

**Se cere realizarea unei functii definite de utilizator, care ar gasi cel mai sarguincios sau cel mai slab student dintr-o grupa. Se defineste urmatorul format al functiei: <nume\_functie> (<Cod\_Grupa>, <is\_good>). Parametrul <is\_good> poate accepta valorile "sarguincios" sau "slab", respectiv. Functia sa returneze un tabel cu urmatoarele campuri Grupa, Nume\_Prenume\_Student, Nota Medie , is\_good. Nota Medie sa fie cu precizie de 2 zecimale.**

create function fun9 (@cod\_grupa VARCHAR(10), @is\_good VARCHAR(20))

returns @Student Table (Cod\_Grupa varchar(10), Student varchar (100), Media decimal(4,2), Reusita varchar(20))

as

begin

if @is\_good = 'sarguincios'

begin

insert into @Student

select top (1) Cod\_Grupa, concat(Nume\_Student,' ',Prenume\_Student) as Student,

CAST(AVG( Nota \* 1.0) as decimal (4,2)) as Media, @is\_good

from grupe,s\_s, rs\_s

where grupe.Id\_Grupa = rs\_s.Id\_Grupa

AND s\_s.Id\_Student = rs\_s.Id\_Student

AND Cod\_Grupa = @cod\_grupa

group by Cod\_Grupa, Nume\_Student, Prenume\_Student

order by Media desc

end

else

begin

insert into @Student

select top (1) Cod\_Grupa, concat(Nume\_Student,' ',Prenume\_Student) as Student,

CAST(AVG( Nota \* 1.0) as decimal (4,2)) as Media, @is\_good

from grupe,s\_s, rs\_s

where grupe.Id\_Grupa = rs\_s.Id\_Grupa

AND s\_s.Id\_Student = rs\_s.Id\_Student

AND Cod\_Grupa = @cod\_grupa

group by Cod\_Grupa, Nume\_Student, Prenume\_Student

order by Media

end

RETURN

end

go

select \* from fun9 ('INF171','sarguincios')

go

select \* from fun9 ('INF171','slab')

