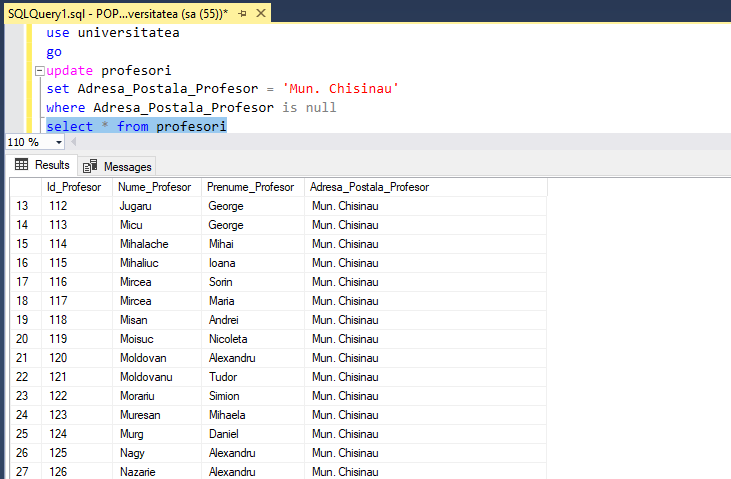
**Lab 6 : CREAREA TABELELOR SI INDEX-ILOR**

**TASK 1 Sa se scrie o instructiune T-SQL, care ar popula coloana Adresa \_ Postala \_ Profesor din tabelul profesori cu valoarea 'Mun. Chisinau', unde adresa este necunoscuta:**

update profesori

set Adresa\_Postala\_Profesor = 'Mun. Chisinau'

where Adresa\_Postala\_Profesor is null

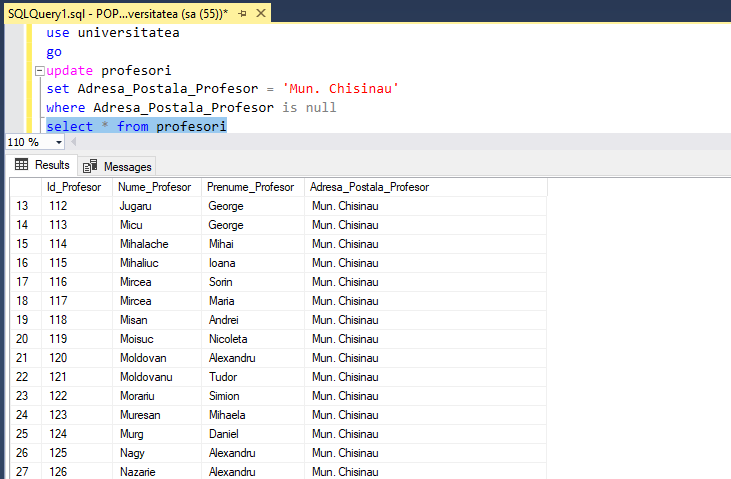


**TASK 2 Sa se modifice schema tabelului grupe, ca sa corespunda urmatoarelor cerinte: a) Campul Cod\_ Grupa sa accepte numai valorile unice si sa nu accepte valori necunoscute. b) Sa se tina cont ca cheie primarii, deja, este definitii asupra coloanei Id\_ Grupa.**

create unique index idx\_cod\_grupa

on grupe (cod\_grupa);

exec sp\_helpindex grupe;



**TASK 3 La tabelul "grupe", sa se adauge 2 coloane noi "Sef\_grupa" si "Prof\_Indrumator", ambele de tip INT. Sa se populeze campurile nou-create cu cele mai potrivite candidaturi in baza criteriilor de mai jos: a) "Seful grupei" trebuie sa aiba cea mai buna reusita (medie) din grupa la toate formele de evaluare si la toate disciplinele. Un student nu poate fi sef de grupa la mai multe grupe. b) Profesorul indrumator trebuie sa predea un numar maximal posibil de discipline la grupa data. Daca nu exista o singura candidatura, care corespunde primei cerinte, atunci este ales din grupul de candidati acel cu identificatorul (Id\_Profesor) minimal. Un profesor nu poate fi indrumator la mai multe grupe. c) Sa se scrie instructiunile ALTER, SELECT, UPDATE necesare pentru crearea coloanelor in tabelul "grupe", pentru selectarea candidatilor si inserarea datelor.**

alter table grupe add sef\_grupa int, prof\_indrumator int;

declare @nr\_grupe int = (select count(Id\_Grupa) from grupe)

declare @initial int = 1;

while (@initial <= @nr\_grupe)

begin

update grupe

set sef\_grupa = (select top 1 sel.Id\_Student

from (select Id\_Student, avg(cast(Nota as float)) as Media

from studenti\_reusita

where Id\_Grupa = @initial

group by Id\_Student) sel

order by sel.Media desc),

prof\_indrumator = (select les.Id\_Profesor

from(select top 1 Id\_Profesor, count(distinct Id\_Disciplina) as Nr\_discipline

from studenti\_reusita

where Id\_Grupa = @initial

group by Id\_Profesor

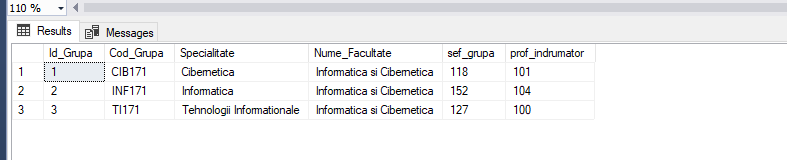
order by Nr\_discipline desc) les)

where Id\_Grupa = @initial

set @initial = @initial +1;

end

alter table grupe add constraint prof\_stud unique(sef\_grupa,prof\_indrumator);



**TASK 4 Sa se scrie o instructiune T-SQL, care ar mari toate notele de evaluare a sefilor de grupe cu un punct. Nota maximala (10) nu poate fi marita:**

update studenti\_reusita

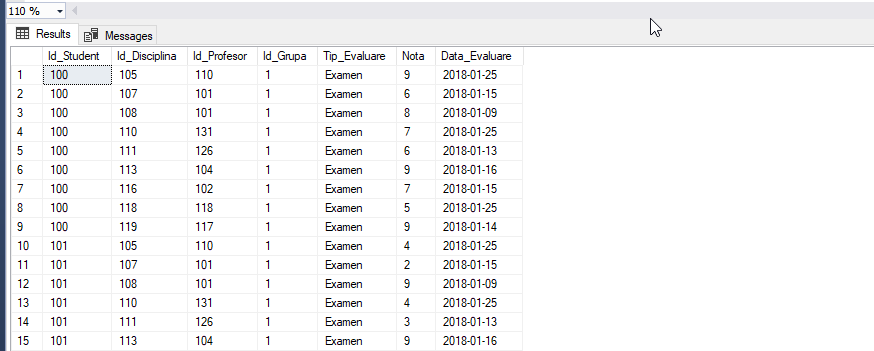
set Nota = Nota + 1

where Tip\_Evaluare = 'Examen'

and Id\_Student = any(select sef\_grupa

from grupe)

and Nota < 10



**TASK 5 Sa se creeze un tabel profesori\_new, care include urmatoarele coloane: Id\_Profesor,Nume \_ Profesor, Prenume \_ Profesor, Localitate, Adresa \_ 1, Adresa \_ 2.**

5.A: Coloana Id\_Profesor trebuie sa fie definita drept cheie primara și, în baza ei, sa fie construit un index CLUSTERED.

5.B: Cîmpul Localitate trebuie sa posede proprietatea DEFAULT= 'mun. Chisinau'.

5.C: Să se insereze toate datele din tabelul profesori în tabelul profesori\_new. Să se scrie, cu acest scop, un număr potrivit de instrucțiuni T-SQL.

În coloana Localitate să fie inserata doar informatia despre denumirea localității din coloana-sursă Adresa\_Postala\_Profesor. În coloana Adresa\_l, doar denumirea străzii. În coloana Adresa\_2, să se păstreze numărul casei și (posibil) a apartamentului.

CREATE TABLE profesori\_new

(

Id\_Profesor int NOT NULL,

Nume\_Profesor char (50),

Prenume\_Profesor char (50),

Localitate char (60) DEFAULT ('mun. Chisinau'),

Adresa\_1 char (60),

Adresa\_2 char (60),

CONSTRAINT [PK\_profesori\_new] PRIMARY KEY CLUSTERED (Id\_Profesor )

) ON [PRIMARY]

INSERT INTO profesori\_new (Id\_Profesor,Nume\_Profesor, Prenume\_Profesor, Localitate,Adresa\_1, Adresa\_2)

(SELECT Id\_Profesor, Nume\_Profesor, Prenume\_Profesor, Adresa\_Postala\_Profesor, Adresa\_Postala\_Profesor, Adresa\_Postala\_Profesor

from profesori)

UPDATE profesori\_new

SET Localitate = case when CHARINDEX(', s.',Localitate) >0

then case when CHARINDEX (', str.',Localitate) > 0

then SUBSTRING (Localitate,1, CHARINDEX (', str.',Localitate)-1)

when CHARINDEX (', bd.',Localitate) > 0

then SUBSTRING (Localitate,1, CHARINDEX (', bd.',Localitate)-1)

end

when CHARINDEX(', or.',Localitate) >0

then case when CHARINDEX (', str.',Localitate) > 0

then SUBSTRING (Localitate,1, CHARINDEX ('str.',Localitate)-3)

when CHARINDEX (', bd.',Localitate) > 0

then SUBSTRING (Localitate,1, CHARINDEX ('bd.',Localitate)-3)

end

when CHARINDEX('nau',Localitate) >0

then SUBSTRING(Localitate, 1, CHARINDEX('nau',Localitate)+2)

end

UPDATE profesori\_new

SET Adresa\_1 = case when CHARINDEX('str.', Adresa\_1)>0

then SUBSTRING(Adresa\_1,CHARINDEX('str',Adresa\_1), PATINDEX('%, [0-9]%',Adresa\_1)- CHARINDEX('str.',Adresa\_1))

when CHARINDEX('bd.',Adresa\_1)>0

then SUBSTRING(Adresa\_1,CHARINDEX('bd',Adresa\_1), PATINDEX('%, [0-9]%',Adresa\_1) - CHARINDEX('bd.',Adresa\_1))

end

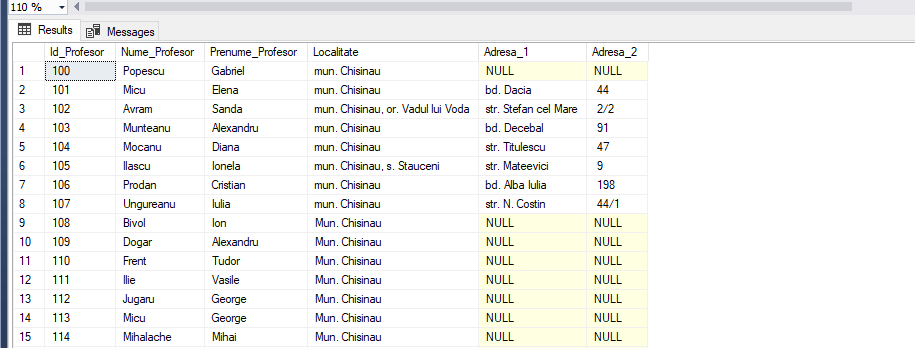
UPDATE profesori\_new

SET Adresa\_2 = case when PATINDEX('%, [0-9]%',Adresa\_2)>0

then SUBSTRING(Adresa\_2, PATINDEX('%, [0-9]%',Adresa\_2)+1,len(Adresa\_2) - PATINDEX('%, [0-9]%',Adresa\_2)+1)

end

select \* from profesori\_new



**TASK 6 Să se insereze datele in tabelul orarul pentru Grupa= 'CIBJ 71' (Id\_ Grupa= 1) pentru ziua de luni. Toate lectiile vor avea loc în blocul de studii 'B'. Mai jos, sunt prezentate detaliile de inserare:**

(ld\_Disciplina = 107, Id\_Profesor= 101, Ora ='08:00', Auditoriu = 202);

(Id\_Disciplina = 108, Id\_Profesor= 101, Ora ='11:30', Auditoriu = 501);

(ld\_Disciplina = 119, Id\_Profesor= 117, Ora ='13:00', Auditoriu = 501);

create table orarul

(

Id\_Disciplina int,

Id\_Profesor int,

Id\_Grupa smallint default(1),

Zi char(10),

Ora Time,

Auditoriu int,

Bloc char(1) default('B'),

PRIMARY KEY (Id\_Grupa, Zi, Ora, Auditoriu)

)

Insert orarul (Id\_Disciplina , Id\_Profesor, Zi, Ora, Auditoriu)

values ( 107, 101, 'Luni','08:00', 202 )

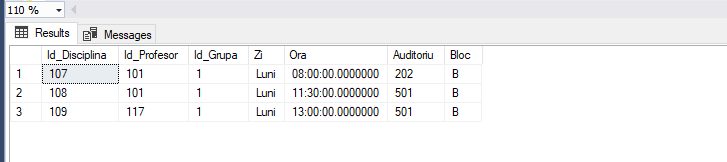
Insert orarul (Id\_Disciplina , Id\_Profesor, Zi, Ora, Auditoriu)

values ( 108, 101, 'Luni','11:30', 501 )

Insert orarul (Id\_Disciplina , Id\_Profesor, Zi, Ora, Auditoriu)

values ( 109, 117, 'Luni','13:00', 501 )

select \* from orarul



**TASK 7 Sa se scrie expresiile T-SQL necesare pentru a popula tabelul orarul pentru grupa INF171 , ziua de luni. Datele necesare pentru inserare trebuie sa fie colectate cu ajutorul instructiunii/instructiunilor SELECT si introduse in tabelul-destinatie, stiind ca:**

lectie #1 (Ora ='08:00', Disciplina = 'Structuri de date si algoritmi', Profesor ='Bivol Ion')

lectie #2 (Ora ='11 :30', Disciplina = 'Programe aplicative', Profesor ='Mircea Sorin')

lectie #3 (Ora ='13:00', Disciplina ='Baze de date', Profesor = 'Micu Elena')

INSERT INTO orarul (Id\_Disciplina, Id\_Profesor, Id\_Grupa, Zi, Ora)

values (

(select Id\_Disciplina from discipline where Disciplina = 'Structuri de date si algoritmi'),

(select Id\_Profesor from profesori where Nume\_Profesor = 'Bivol' and Prenume\_Profesor = 'Ion'),

(select Id\_Grupa from grupe where Cod\_Grupa = 'INF171'),

'Luni',

'08:00')

INSERT INTO orarul (Id\_Disciplina, Id\_Profesor, Id\_Grupa, Zi, Ora)

values (

(select Id\_Disciplina from discipline where Disciplina = 'Programe aplicative'),

(select Id\_Profesor from profesori where Nume\_Profesor = 'Mircea' and Prenume\_Profesor = 'Sorin'),

(select Id\_Grupa from grupe where Cod\_Grupa = 'INF171'),

'Luni',

'11:30')

INSERT INTO orarul (Id\_Disciplina, Id\_Profesor, Id\_Grupa, Zi, Ora)

values (

(select Id\_Disciplina from discipline where Disciplina = 'Baze de date'),

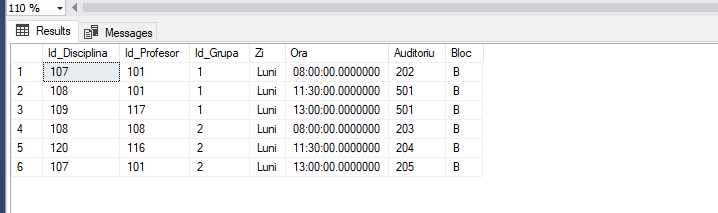
(select Id\_Profesor from profesori where Nume\_Profesor = 'Micu' and Prenume\_Profesor = 'Elena'),

(select Id\_Grupa from grupe where Cod\_Grupa = 'INF171'),

'Luni',

'13:00')

select \* from orarul



**TASK 8 Sa se scrie interogarile de creare a indecsilor asupra tabelelor din baza de date universitatea pentru a asigura o performanta sporita la executarea interogarilor SELECT din Lucrarea practica 4. Rezultatele optimizarii sa fie analizate in baza planurilor de executie, pana la si dupa crearea indecsilor. Indecsii nou-creati sa fie plasati fizic in grupul de fisiere userdatafgroupl (Crearea si intrefinerea bazei de date - sectiunea 2.2.2)**

SELECT \*

INTO studenti\_reusita\_test

FROM studenti\_reusita

create CLUSTERED INDEX [VX\_Test]

ON studenti\_reusita\_test (Id\_Disciplina, Id\_Profesor, Id\_Grupa)

UPDATE studenti\_reusita\_test

set Nota = Nota + 1

where Id\_Student = ANY (select Sef\_grupa from grupe)

and Nota < 10

select Sef\_grupa, Nota, Tip\_Evaluare , Id\_Disciplina

from grupe , studenti\_reusita\_test

where grupe.Id\_Grupa = studenti\_reusita\_test.Id\_Grupa

