Student: Călin Andreea Cleopatra

**PROIECT-TEMA 11**

1. Crearea bazei de date a fost realizata conform schemei primite. Drept chei primare am folosit NR\_LEGITIMATIE din tabelul student si CNP din tabelul profesor. Pe acestea le-am folosit in continuare drept foreign keys in restul tabelelor.

**Crearea tabelelor si inserarea datelor** (partea de insert din documentatie uneori va contine doar o parte din adaugari ):

* **Student:**

drop table student;

create table student(

NR\_LEGITIMATIE integer not null,

NUME varchar(20),

PRENUME varchar(20),

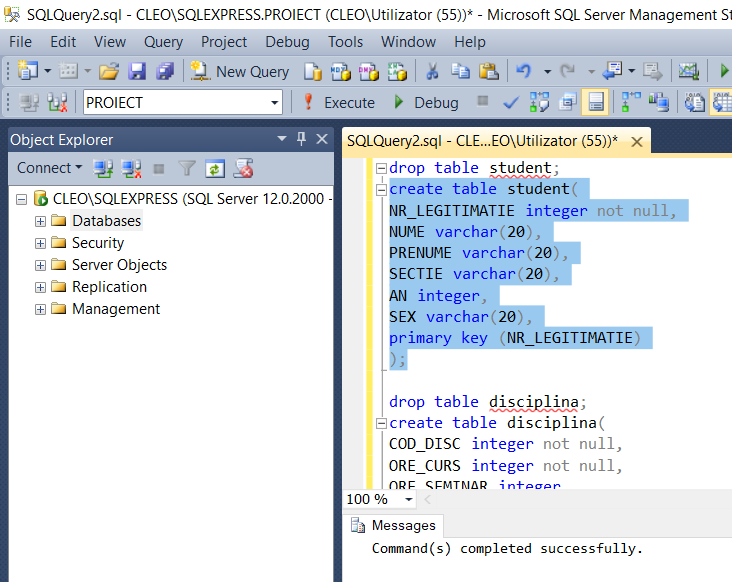
SECTIE varchar(20),

AN integer,

SEX varchar(20),

primary key (NR\_LEGITIMATIE)

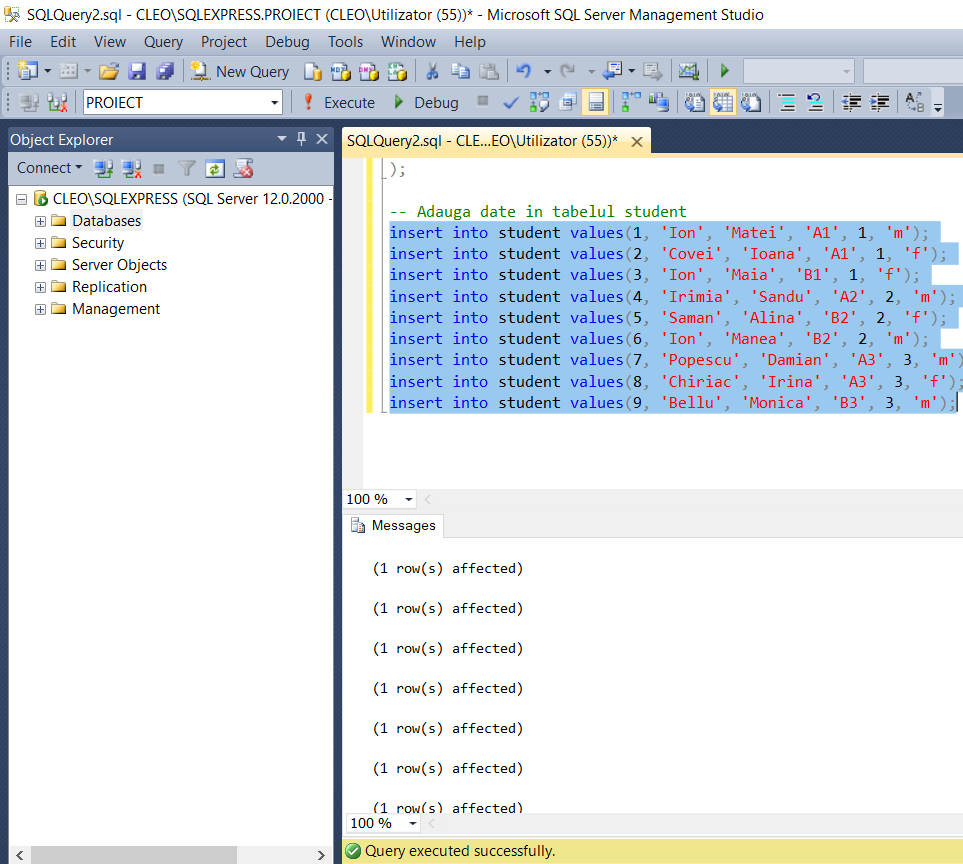
);



-- Adauga date in tabelul student

insert into student values(1, 'Ion', 'Matei', 'A1', 1, 'm');

insert into student values(2, 'Covei', 'Ioana', 'A1', 1, 'f');

insert into student values(3, 'Ion', 'Maia', 'B1', 1, 'f');

* **Disciplina:**

drop table disciplina;

create table disciplina(

COD\_DISC integer not null,

ORE\_CURS integer not null,

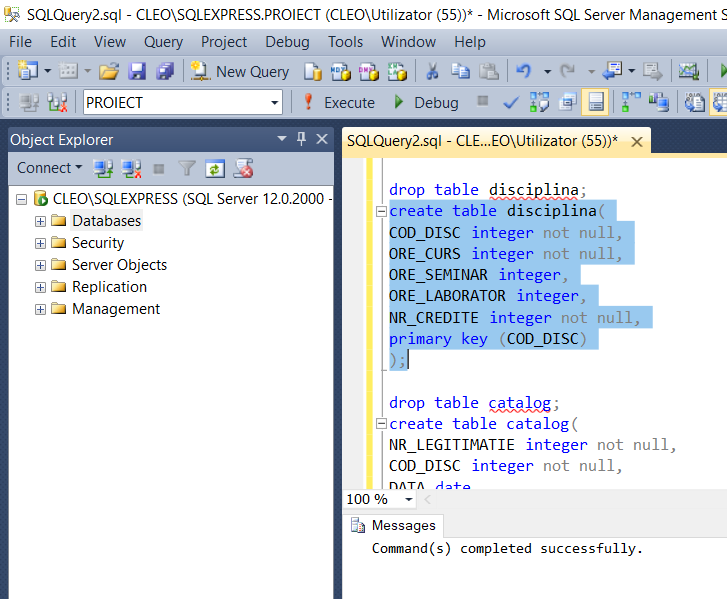
ORE\_SEMINAR integer,

ORE\_LABORATOR integer,

NR\_CREDITE integer not null,

primary key (COD\_DISC)

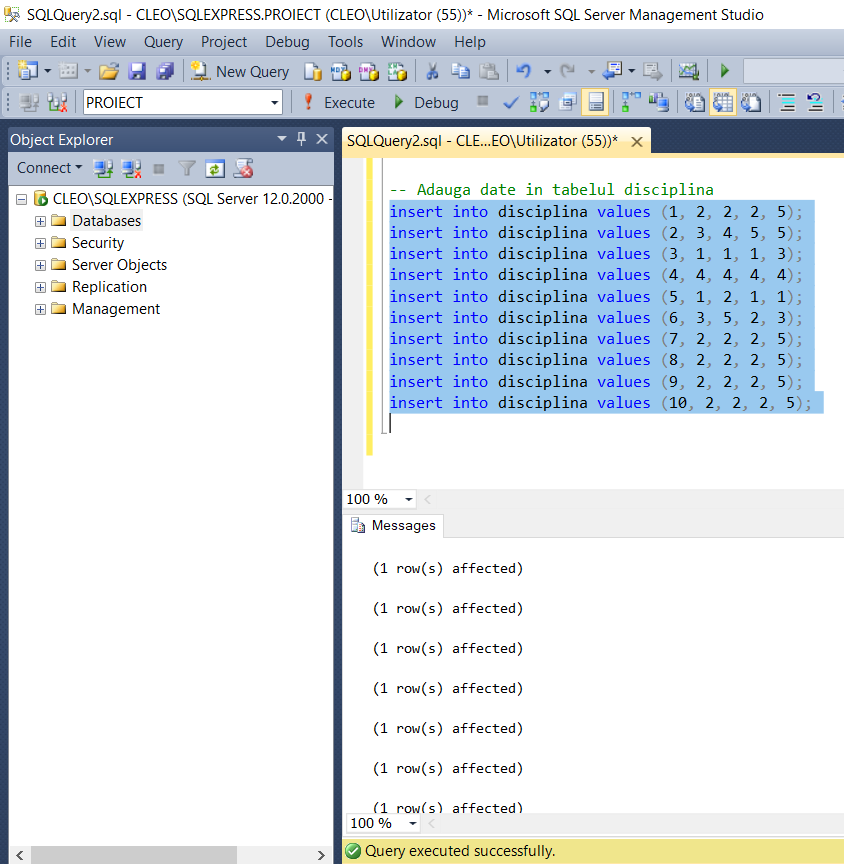
);

****

-- Adauga date in tabelul disciplina

insert into disciplina values (1, 2, 2, 2, 5);

insert into disciplina values (2, 3, 4, 5, 5);

****insert into disciplina values (3, 1, 1, 1, 3);

* **Catalog:**

drop table catalog;

create table catalog(

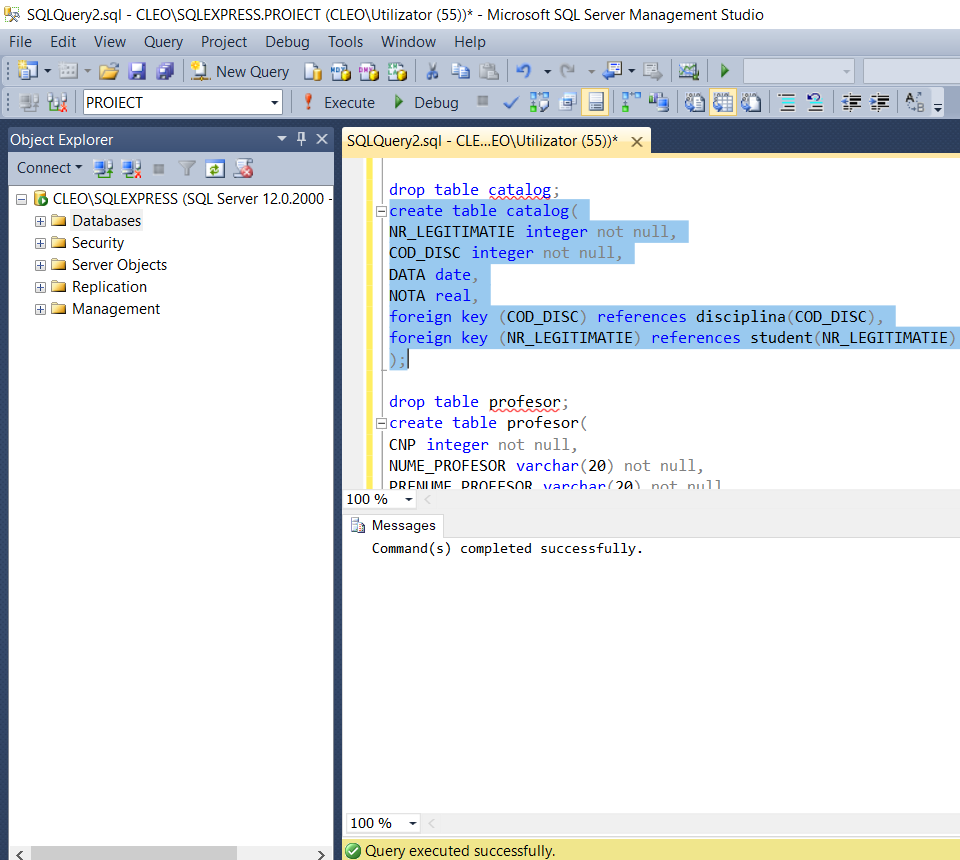
NR\_LEGITIMATIE integer not null,

COD\_DISC integer not null,

DATA date,

NOTA real,

foreign key (COD\_DISC) references disciplina(COD\_DISC),

****foreign key (NR\_LEGITIMATIE) references student(NR\_LEGITIMATIE));

-- Adauga date in tabelul catalog

insert into catalog values(1, 1, '2008-7-04', 10);

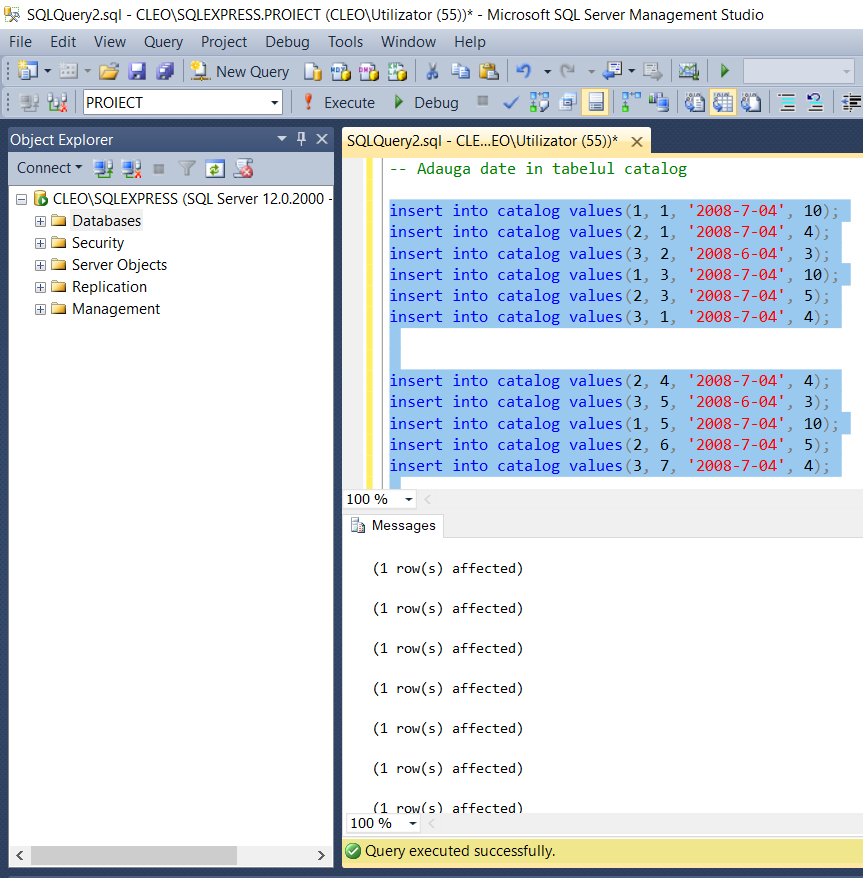
insert into catalog values(2, 1, '2008-7-04', 4);

insert into catalog values(3, 2, '2008-6-04', 3);

insert into catalog values(1, 3, '2008-7-04', 10);

insert into catalog values(2, 3, '2008-7-04', 5);

insert into catalog values(3, 1, '2008-7-04', 4);

****

* **Profesor:**

drop table profesor;

create table profesor(

CNP integer not null,

NUME\_PROFESOR varchar(20) not null,

PRENUME\_PROFESOR varchar(20) not null,

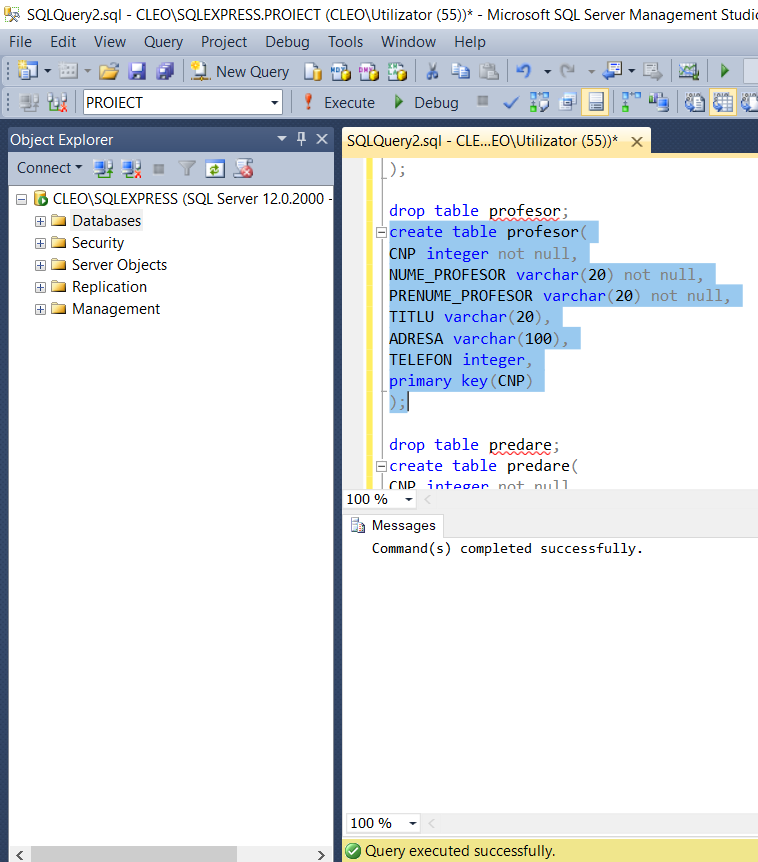
TITLU varchar(20),

ADRESA varchar(100),

TELEFON integer,

primary key(CNP)

);

****

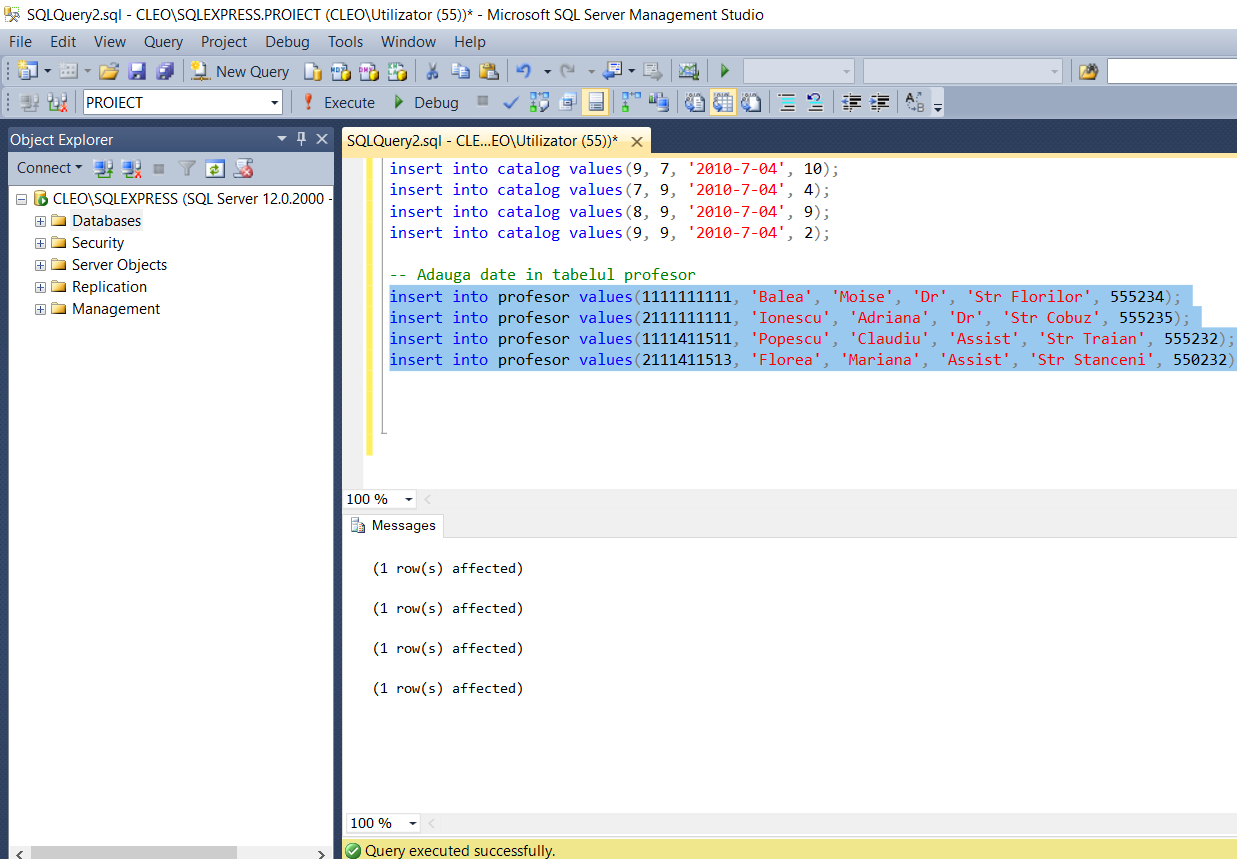
-- Adauga date in tabelul profesor

insert into profesor values(1111111111, 'Balea', 'Moise', 'Dr', 'Str Florilor', 555234);

insert into profesor values(2111111111, 'Ionescu', 'Adriana', 'Dr', 'Str Cobuz', 555235);

insert into profesor values(1111411511, 'Popescu', 'Claudiu', 'Assist', 'Str Traian', 555232);

insert into profesor values(2111411513, 'Florea', 'Mariana', 'Assist', 'Str Stanceni', 550232);



* **Predare:**

drop table predare;

create table predare(

CNP integer not null,

COD\_DISC integer not null,

CURS integer not null,

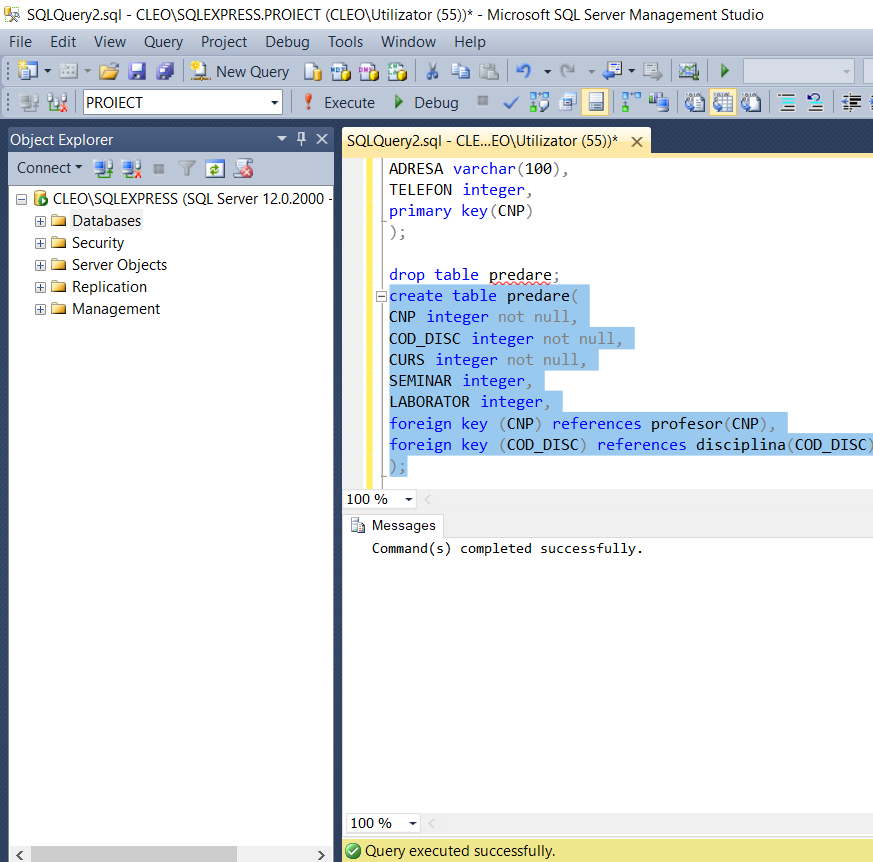
SEMINAR integer,

LABORATOR integer,

foreign key (CNP) references profesor(CNP),

foreign key (COD\_DISC) references disciplina(COD\_DISC)

);

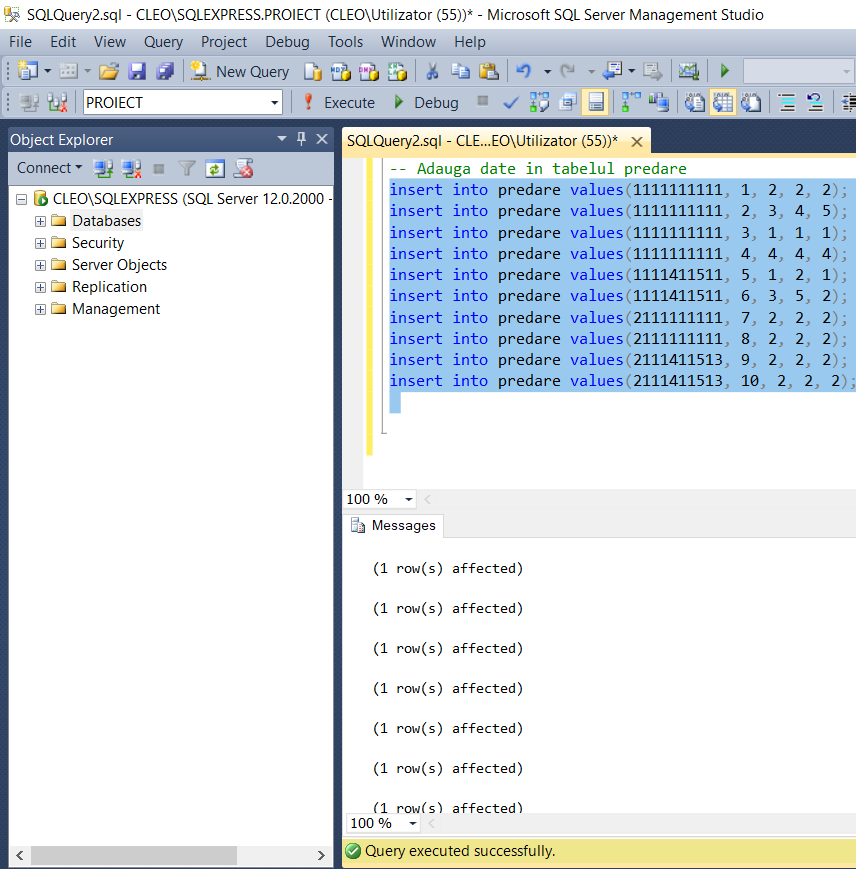
****

-- Adauga date in tabelul predare

insert into predare values(1111111111, 1, 2, 2, 2);

insert into predare values(1111111111, 2, 3, 4, 5);

insert into predare values(1111111111, 3, 1, 1, 1);



1. Pentru exercitiul 2 am utilizat intrarile din catalog si am numarat aparitiile NR\_LEGITIMATIE (prin functia count) iar gruparea am facut-o dupa COD\_DISC.

* Determinati studentii care s-au prezentat la o disciplina de mai mult de 2 ori.

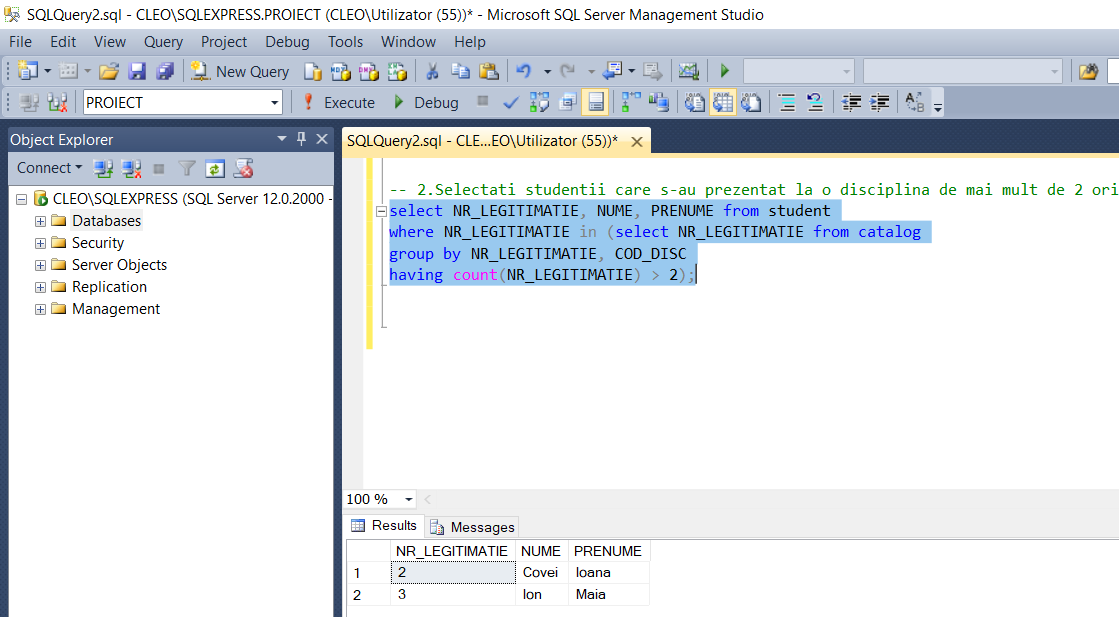
select NR\_LEGITIMATIE, NUME, PRENUME from student

where NR\_LEGITIMATIE in (select NR\_LEGITIMATIE from catalog

group by NR\_LEGITIMATIE, COD\_DISC

having count(NR\_LEGITIMATIE) > 2);

Studentii ce au fost selectati sunt cei cu numarul legitimatiei 2 (Covei Ioana) si numarul legitimatiei 3 (Ion Maia).



1. Pentru exercitiul 3 am procedat similar exercitiului doi, diferenta fiind ca in loc de a face numarare prin functia count, am calculat media studentilor prin functia avg si am comparat-o cu 8, grupand notele dupa NR\_LEGITIMATIE.

* Determinati studentii care au media > 8 in fiecare an

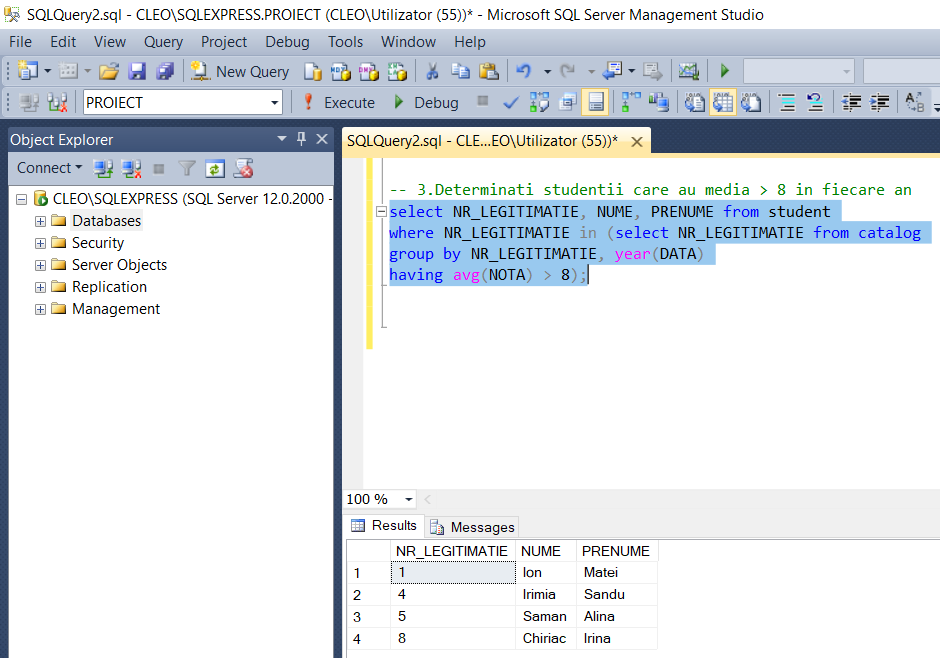
select NR\_LEGITIMATIE, NUME, PRENUME from student

where NR\_LEGITIMATIE in (select NR\_LEGITIMATIE from catalog

group by NR\_LEGITIMATIE, year(DATA)

having avg(NOTA) > 8);

Studentii ce au fost selectati sunt cei cu numarul legitimatiei 1(Ion Matei), 4(Irimia Sandu), 5(Saman Alina), 8(Chiriac Irina).



1. La exercitiul 4 pentru a crea procedura a trebuit sa folosesc case pentru a face suma dupa NR\_CREDITE doar pentru materiile la care studentul are mai mult de 5. Am facut join intre tabelele student, catalog si disciplina pentru a lua informatii despre studenti, note si numarul de credite la fiecare materie.

* Crearea procedurii de calcul.

CREATE PROCEDURE Procedura

@Credite int

AS

select s.NR\_LEGITIMATIE, sum(case when c.NOTA >= 5 then d.NR\_CREDITE else 0 end) as Credits from student s

left join catalog c on s.NR\_LEGITIMATIE=c.NR\_LEGITIMATIE

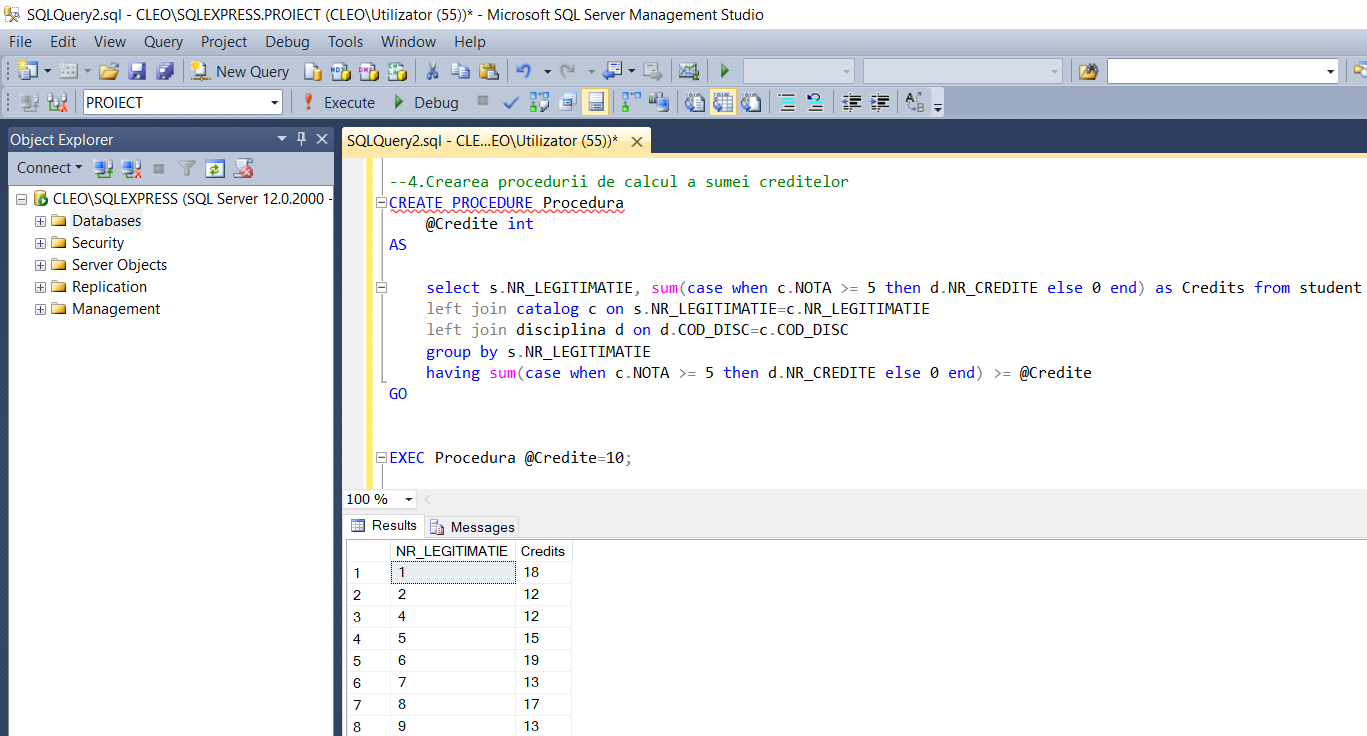
left join disciplina d on d.COD\_DISC=c.COD\_DISC

group by s.NR\_LEGITIMATIE

having sum(case when c.NOTA >= 5 then d.NR\_CREDITE else 0 end) >= @Credite

GO

Studentii selectati sunt cei cu numarul legitimatiei 1(18 credite), 2(12 credite), 4(12 credite), 5(15 credite), 6(19 credite), 7(13 credite), 8(17 credite) si 9(13 credite).



1. La exercitiul 5 pentru a gasi materiile cu cei mai multi restantieri a trebuit sa calculez media fiecarui student la fiecare materie. Acest lucru a fost posibil prin aplicarea functiei avg, impreuna cu gruparea dupa NR\_LEGITIMATIE (particular pentru fiecare student trebuie calculata media), COD\_DISC (ca sa observ daca nu cumva au restante la mai multe discipline) si NOTA (pentru a filtra notele in medie). Pentru a afisa primele inregistrari am folosit top.

* Disciplinele cu cei mai multi restantieri.

select distinct top 2 d.COD\_DISC, count(\*) from disciplina d

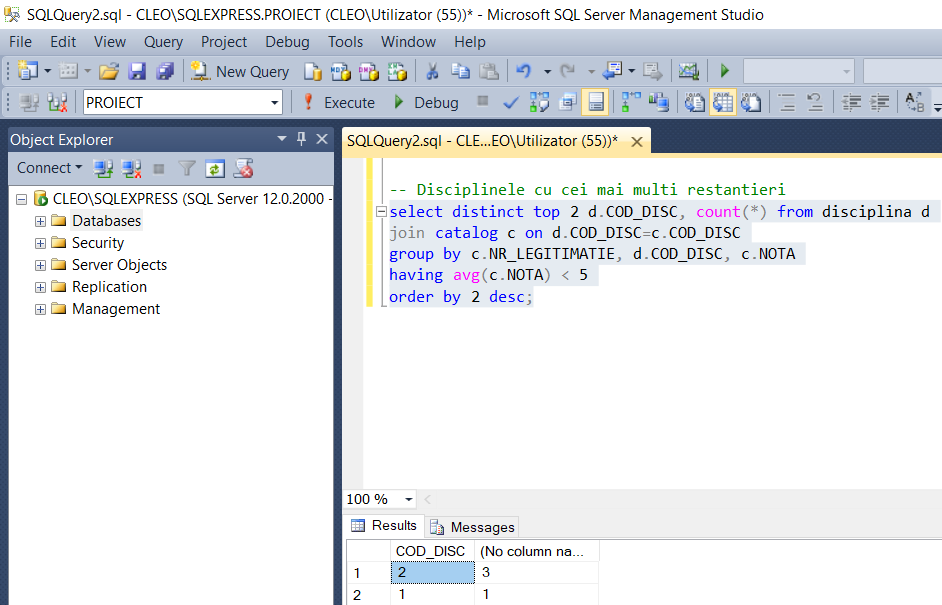
join catalog c on d.COD\_DISC=c.COD\_DISC

group by c.NR\_LEGITIMATIE, d.COD\_DISC, c.NOTA

having avg(c.NOTA) < 5

order by 2 desc;

Disciplinele selectate sunt cele cu codul 2 (3 restante) si 1(o restanta).



1. La exercitiul 6 am construit un view care afiseaza notele studentilor grupate dupa AN si SECTIE. Aici am facut join intre student si catalog pentru a putea lua datele personale ale studentilor (nume, an, sectie) si notele primite.

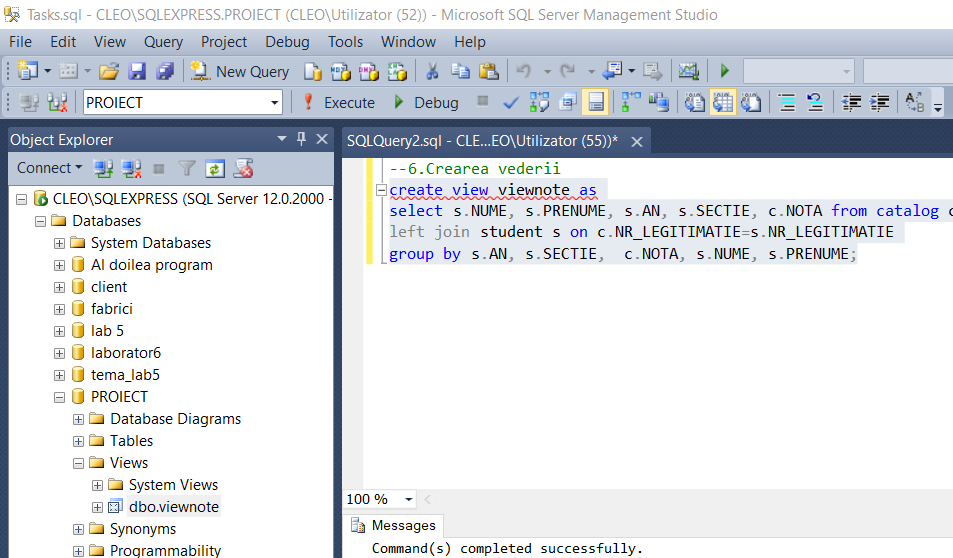
* Crearea vederii.

create view viewnote as

select s.NUME, s.PRENUME, s.AN, s.SECTIE, c.NOTA from catalog c

left join student s on c.NR\_LEGITIMATIE=s.NR\_LEGITIMATIE

group by s.AN, s.SECTIE, c.NOTA, s.NUME, s.PRENUME;



1. La exercitiul 7 pentru vizualizarea profesorilor cu cei mai multi restantieri am facut similar cu punctul 6, numai ca a trebuit sa iau informatii din 4 tabele: profesor, predare, catalog si disciplina. Gruparea s-a facut dupa datele personale ale profesorului.

* Profesorii cu cei mai multi restantieri.

select top 2 p.CNP, p.NUME\_PROFESOR, p.PRENUME\_PROFESOR, count(distinct c.NR\_LEGITIMATIE) from disciplina d

join catalog c on d.COD\_DISC=c.COD\_DISC

join predare pr on pr.COD\_DISC=c.COD\_DISC

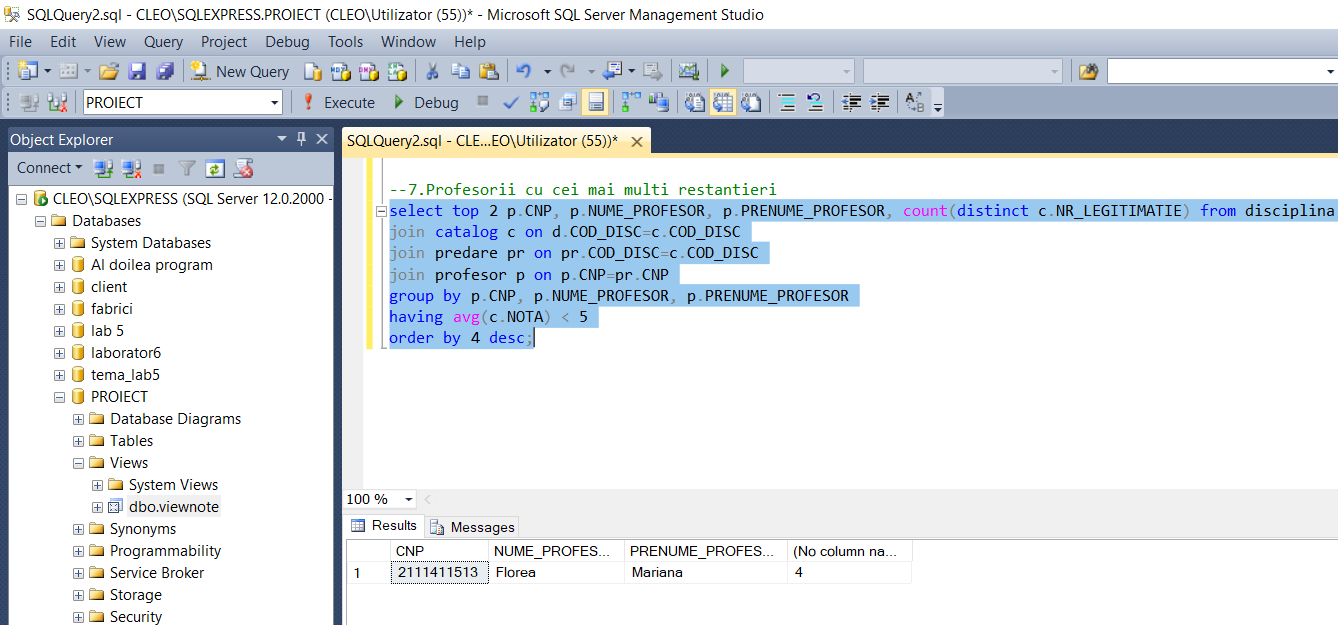
join profesor p on p.CNP=pr.CNP

group by p.CNP, p.NUME\_PROFESOR, p.PRENUME\_PROFESOR

having avg(c.NOTA) < 5

order by 4 desc;

Profesorul selectat este Florea Mariana.



1. La exericitiul 8 pentru a gasi studentii care au mai mult de 4 examene nepromovate am facut join intre tabelele catalog si student pentru a accesa informatiile despre studenti, grupate in functie de AN,SECTIE, NOTA, NUME si PRENUME. Vor fi selectati prin functia having studentii ce au nota mai mica decat 5 la mai mult de 4 examene (numarul examenelor este contorizat prin functia count).

* Studentii cu mai mult de 4 examene nepromovate.

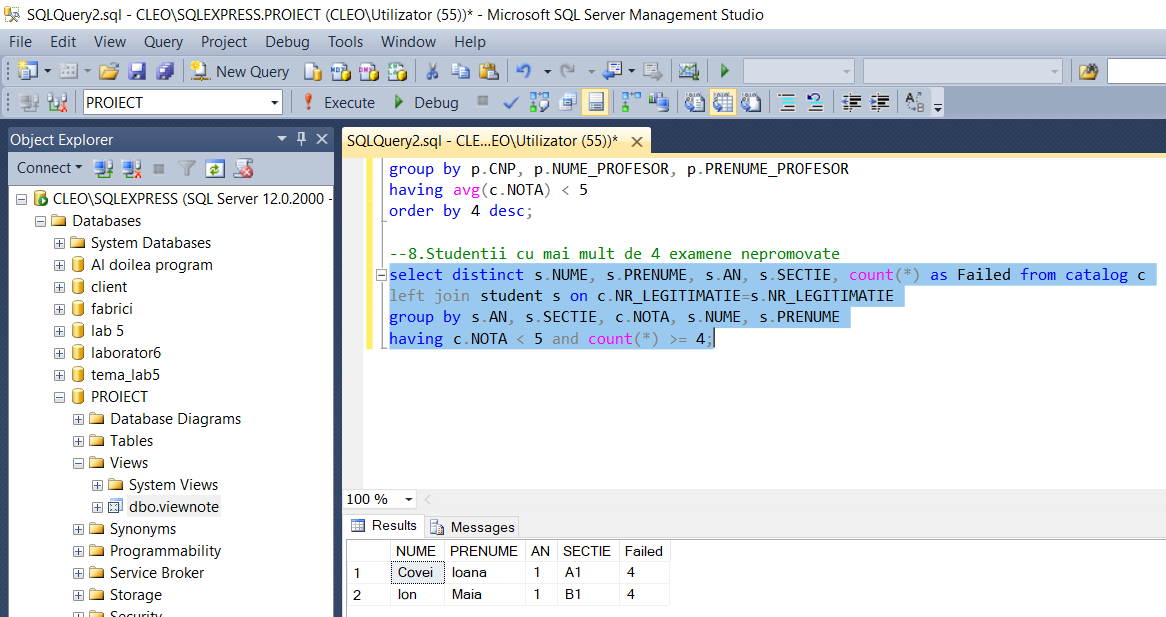
select distinct s.NUME, s.PRENUME, s.AN, s.SECTIE, count(\*) as Failed from catalog c

left join student s on c.NR\_LEGITIMATIE=s.NR\_LEGITIMATIE

group by s.AN, s.SECTIE, c.NOTA, s.NUME, s.PRENUME

having c.NOTA < 5 and count(\*) >=4;

Studentii selectati sunt Covei Ioana si Ion Maia.



1. La exercitiul 9 pentru realizarea diagramei bazei de date am folosit dbdiagram.io. In figura se pot observa cheile primare si legaturile intre tabele.

