

Universitatea din București
Facultatea de Matematică și Informatică

PROIECT BAZE DE DATE MANAGEMENTUL UNEI FACULTĂȚI

Profesor coordonator:
Vasile Silviu Laurențiu

Student:
Grigoruță Cosmin-Ioan

București, 2020

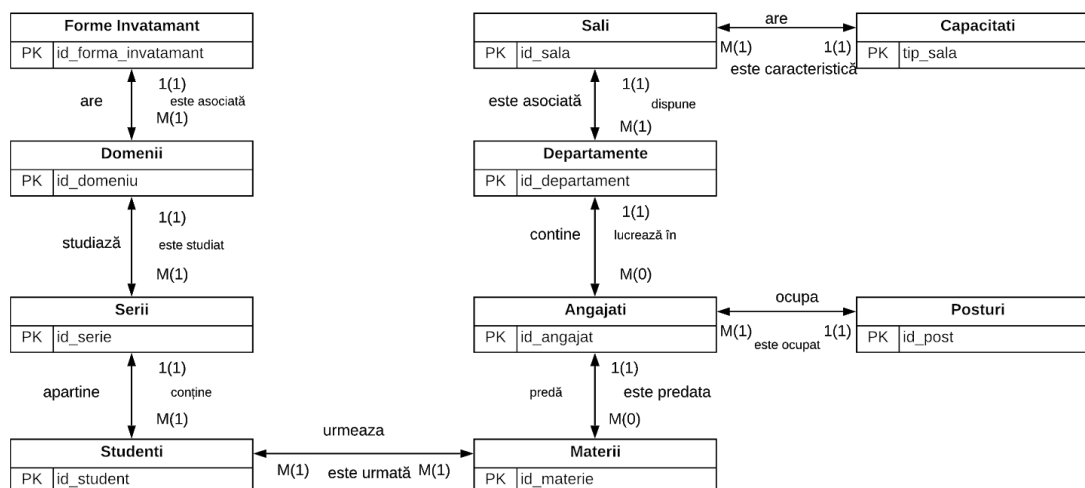
Cuprins

1. Prezentarea modelului și a regulilor acestuia
2. Diagrama entitate-relație
3. Diagrama conceptuală
4. Crearea tabelelor
5. Introducerea Datelor

1. Prezentarea modelului și a regulilor acestuia

Proiectul are ca scop gestionarea activităților unei facultăți precum organizarea ciclurilor universitare de studii de care aceasta dispune, managementul studenților și al angajaților, dar și al sălilor unde se desfășoară procesul de învățare și cel organizatoric. Perspectiva modelului se concentrează asupra componentei umane și a activităților desfășurate de aceasta. Sunt înregistrate date cu caracter personal despre studenți și angajați, în același timp aceștia sunt asociați unor structuri organizatorice (serii, respectiv departamente și posturi). La rândul lor, acestor structuri le sunt atribuite caracteristici de diferențiere (domenii de studii, forme de învățământ) în cazul activităților de studiu, în cazul activităților organizatorice este atribuită componenta logistică de care instituția dispune (săli, tipurile și capacitățile acestora).

2. Diagrama entitate-relație



Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

Entitatea STUDENTI conține informații despre fiecare student ce aparține facultății respective, sunt incluse datele personale precum numele, prenumele, data de naștere, sexul, naționalitate - acest câmp are rolul de a oferi informații în cazul în care se doresc statistici despre numărul studenților internaționali sau cei care studiază un număr mai mic de semestre prin intermediul programelor de tipul ERASMUS. În același timp sunt păstrate informații despre statutul finanțării studentului prin intermediul câmpului “taxa”.

Entitatea SERII a fost creată cu scopul de a grupa studenții în funcție de domeniul pe care l-au optat. Aceasta conține informații despre data în care aceștia ar trebui să finalizeze studiile, pot fi obținute date despre numărul studenților ce vor absolvi ciclul respectiv de studii universitare în anumiți ani.

Entitatea DOMENII are rolul de a gestiona informațiile despre posibilitățile de studiu pe care facultatea le oferă, precum și numărul de ani necesari finalizării acestuia.

Entitatea FORME INVATAMANT pune la dispoziție informații suplimentare despre domeniile de studiu oferite întrucât pot fi diferite opțiuni în privința urmăririi domeniului. Formele de învățământ diferă prin metodele de predare și învățare.

Entitatea MATERII oferă date precum nume, număr de credite și forma de examinare la cursurile urmate la facultatea respectivă.

Entitatea ANGAJATI conține informații precum nume, prenume, statutul pe care acesta îl deține în cadrul facultății împreună cu datele de contact și venitul acestuia.

Entitatea DEPARTAMENTE are rolul de a organiza angajații în funcție de rolul pe care îl ocupă din punct de vedere al activității educaționale, de cercetare sau administrativă.

Entitatea POSTURI are rolul de a organiza angajații după funcțiile ocupate în activitatea ce o exercită.

Entitatea SALI oferă informații ce vizează componenta logistică de care dispune facultatea în vederea îndeplinirii activităților educaționale, administrative și de cercetare.

Entitatea CAPACITATI a fost creată în vederea reducerii redundanței datelor în tabela anterior menționată, la fel ca aceasta, oferă informații ce vizează componenta logistică.

Aceste entități sunt puse în legătură prin următoarele relații:

Între entitățile SERII și DOMENII există relația - *many to one* - întrucât pot exista mai multe serii de studenți ce studiază în același domeniu.

Între entitățile DOMENII și FORME INVATAMANT există relația - *many to one* - întrucât mai multe domenii de studiu pot fi încadrate în aceeași formă de învățământ.

Între entitățile STUDENTI și SERII există relația - *many to one* - întrucât mai mulți studenți sunt în aceeași serie.

Între entitățile STUDENTI și MATERII există relația - *many to many* - întrucât mai mulți studenți urmează mai multe materii. Această relație *many to many* va fi remediată prin intermediul unui tabel intermediar.

Între entitățile MATERII și ANGAJATI există relația - *many to one* - întrucât un angajat ce ocupă o poziție într-un departament educațional poate preda mai multe materii, dar în același timp un angajat ce ocupă o poziție într-un departament administrativ nu poate preda niciuna, iar o materie poate fi predată de un singur angajat.

Între entitățile ANGAJATI și DEPARTAMENTE există relația - *many to one* - întrucât mai mulți angajați lucrează într-un singur departament.

Între entitățile ANGAJATI și POSTURI există relația - *many to one* - întrucât mai mulți angajați pot ocupa același post.

Între entitățile DEPARTAMENTE și SALI există relația - *one to many* - întrucât mai multe săli sunt asociate unui departament.

Între entitățile SALI și CAPACITATI există relația - *many to one* - întrucât mai multe săli pot fi de un singur tip, deci au aceeași capacitate.

Totodată, toate entitățile în afară de CAPACITATI au ca cheie primară “id_<numele entității la singular>”. În cazul entității CAPACITATI cheia primară este “tip_sala”.

3. Diagrama conceptuală

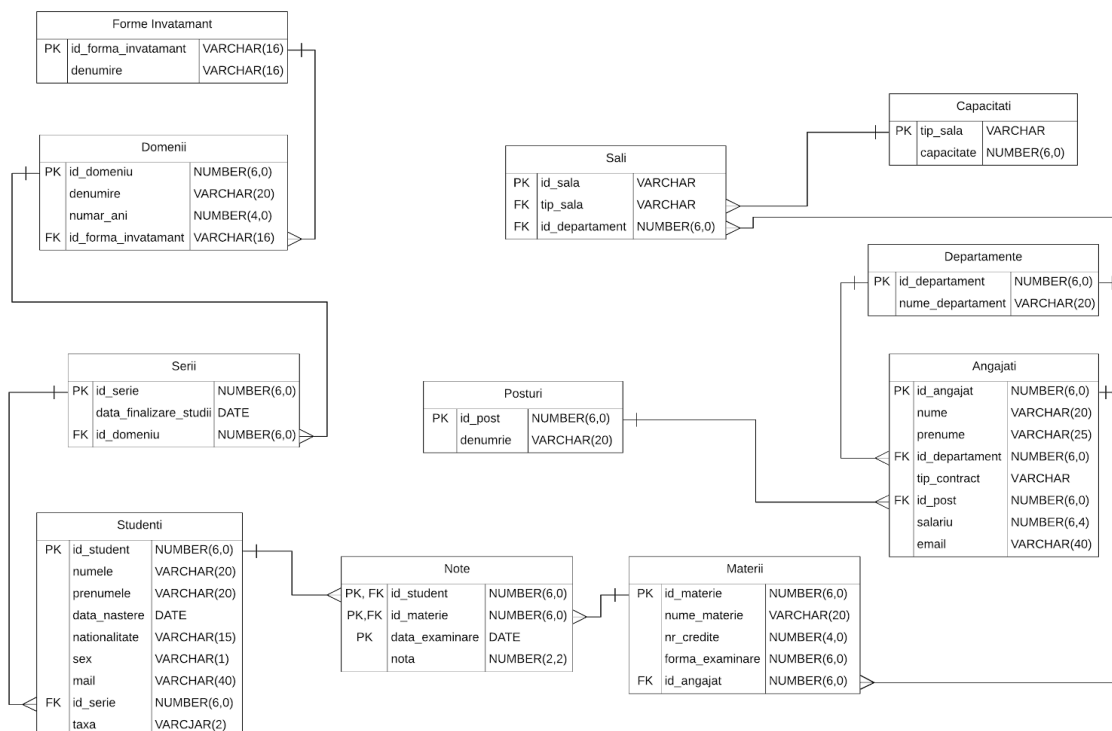


Diagrama conceptuală se diferențiază față de cea entitate-relație prin faptul că entitățile au fost înlocuite de tabele ce conțin atribute caracterizate de tipul de date, au apărut și cheile externe ce compun legăturile cu celelalte tabele, iar cardinalitățile au dispărut.

Tabelul STUDENTI are ca cheie primară câmpul “id_student”, tipul acestuia este “NUMBER(6, 0)”. Acesta are următoarele atribute: “nume” - “VARCHAR(40)”, “prenume” - “VARCHAR(40)”, “data_nastere” - “DATE”, “nationalitate” - “VARCHAR(40)”, “sex” - “VARCHAR(1)”, “email” - “VARCHAR(40)”, “taxa” - “VARCHAR(2)”. Totodată, tabelul prezintă și o cheie externă către tabelul intermediar SERII: “id_serie” - “NUMBER(6, 0)”.

Tabelul SERII are ca cheie primară câmpul “id_serie” cu tipul “NUMBER(6,0)”. Acesta are următoarele câmpuri: “data_finalizare_studii” - “DATE” și cheia externă ce face legătura cu tabelul DOMENII: “id_domeniu” - “NUMBER(6, 0)”.

Tabelul DOMENII are ca cheie primară câmpul “id_domeniu” cu tipul “NUMBER(6,0)”. Acesta are următoarele atribute: “denumire” - “VARCHAR(40)”, “numar_ani” - “NUMBER(6, 0)”, iar cheia externă a acestui tabel este “id_forme_invatamant” - “NUMBER(6,0)” ce face legătura cu tabelul FORME INVATAMANT.

Tabelul FORME INVATAMANT are ca cheie primară câmpul “id_formă_invățământ” cu tipul “NUMBER(6,0)”. Acesta are un singur atribut și anume denumirea completă.

Tabelul ANGAJATI are ca cheie primară câmpul “id_angajat” cu tipul de date “NUMBER(6,0)” ce se va incrementa la adăugarea fiecărei înregistrări. Acesta are următoarele atribute: “nume” - “VARCHAR(40)”, “prenume” - “VARCHAR(25)”, “salariu” - “NUMBER(10,4)”, “email” - “VARCHAR(40)”. În același timp, acest tabel prezintă următoarele chei externe: “id_departament” - “NUMBER(6,0)” ce face legătura cu tabelul DEPARTAMENTE, “id_post” - “NUMBER(6,0)” ce face legătura cu tabelul POSTURI.

Tabelul POSTURI are ca cheie primară câmpul “id_post” cu tipul de date “NUMBER(6,0)”. Acesta are ca atribut câmpul “denumire” - “VARCHAR(40)”.

Tabelul DEPARTAMENTE are ca cheie primară câmpul “id_departament” cu tipul de date “NUMBER(6,0)”. Acesta are ca atribut câmpul “nume_departament” - “VARCHAR(40)”.

Tabelul SALI are ca cheie primară câmpul “id_sala” cu tipul “NUMBER(6,0)”. Acesta prezintă următoarele chei externe: “tip_sala” - “VARCHAR(40)” ce face legătura cu tabelul CAPACITATI și “id_departament” - “NUMBER(6,0)” ce face legătura cu tabelul DEPARTAMENTE.

Tabelul CAPACITATI are ca cheie primară câmpul “tip_sala” cu tipul de date “VARCHAR(40)”. Atributul acestui tabel este “capacitate” - “NUMBER(10)”.

Tabelul MATERII are ca cheie primară câmpul “id_materie” cu tipul de date “NUMBER(6,0)”. Acest tabel prezintă următoarele atribute: “nume_materie” - “VARCHAR(40)”, “numar_credite” - “NUMBER(4,0)”, “forma_examinare” - “VARCHAR(40)”. În același timp, tabela are și o cheie externă “id_angajat” - “NUMBER(6,0)” ce face legătura cu tabelul ANGAJATI.

Tabelul NOTE reprezintă tabelul intermediar menit să rezolve problema relației *many to many* dintre entitățile STUDENTI și MATERIE. Acesta prezintă o cheie primară compusă alcătuită din “id_student”, “id_materie”, “data_examinarii” - “DATE”, primele două chei sunt pentru identificarea evaluării studentului, iar cea din urmă are rolul de a clarifica situația studentului în legătură cu materia. Studentul poate fi evaluat de mai multe ori în cadrul materiei respective, iar data ne permite obținerea unei situații actualizate în timp. Tabelul conține și nota studentului, “nota” - “NUMBER(6,4)”.

Descrierea constrângerilor

În tabelul FORME_INVATAMANT câmpul “denumire” are o constrângere de tipul NOT NULL.

În tabelul DOMENII câmpurile “denumire”, “numar_ani” și “id_forma_invatamant” au o constrângere de tipul NOT NULL, în același timp câmpul “numar_ani” are o constrângere de tip CHECK(numar_ani > 0) întrucât durata unui domeniu de studiu nu poate fi reprezentată de un număr negativ. Totodată, există o constrângere de tip UNIQUE(domeniu, id_forma_invatamant) întrucât să nu poată fi adăugat același domeniu la aceeași formă de învățământ.

În tabelul SERII câmpurile “data_finalizare_studii” și “id_domeniu” prezintă o constrângere de tip NOT NULL deoarece dorim să cunoaștem anul finalizării studiilor pentru studenții ce aparțin domeniului respectiv și totodată seria trebuie asociată unui domeniu.

În tabelul STUDENTI câmpurile “nume”, “prenume”, “data_nastere”, “nationalitate”, “sex”, “email”, “id_serie”, “taxa” au o constrângere de tip NOT NULL. În plus câmpul “sex” are o constrângere de tip CHECK(sex IN (‘M’, ‘F’)), câmpul “taxa” are o constrângere de tip CHECK(taxa IN (‘DA’, ‘NU’)), iar câmpul “email” are o constrângere de tip UNIQUE.

În tabelul NOTE câmpul “nota” are o constrângere de tip CHECK(nota BETWEEN 0 AND 10) întrucât nota unui student la un examen nu poate fi un număr negativ și mai mare decât 10.

În tabelul MATERII câmpul “nume_materie”, “nr_credite”, “forma_examinare” au o constrângere de tipul NOT NULL, în plus câmpul “nr_credite” are o constrângere de tipul CHECK(nr_credite > 0).

În tabelul ANGAJATI câmpurile “nume”, “prenume”, “id_departament”, “salariu”, “email” au o constrângere de tipul NOT NULL. În același timp, câmpul salariu are o constrângere de tip CHECK(salariu > 0) deoarece venitul unui angajat nu poate fi un număr negativ, iar câmpul “email” are o constrângere de tip UNIQUE.

În tabelul POSTURI câmpul “denumire” are o constrângere de tip NOT NULL, este necesară înregistrarea .

În tabelul DEPARTAMENTE câmpul “denumire” are o constrângere de tip NOT NULL și una de tip UNIQUE.

În tabelul SALI există o constrângere de tip UNIQUE(id_sala, tip_sala, id_departament) pentru a nu exista posibilitatea introducerii a unei săli identice în tabelul respectiv.

În tabelul CAPACITATI câmpul “capacitate” are o constrângere de tip CHECK(capacitate > 0) și NOT NULL, în plus la nivelul tabelului mai există o constrângere de tip UNIQUE(tip_sala, capacitate).

Prezentarea schemelor relaționale

Un student nu poate învăța la o facultate fără să fie inclus într-o serie. Datorită acestui fapt, în momentul în care o serie este ștearsă vor fi șterse și informațiile tuturor studenților ce aparțin seriei respective prin intermediul unei constrângeri ON DELETE CASCADE asupra cheii externe “id_serie” din tabelul STUDENTI.

O serie nu poate exista fără să fie inclusă într-un domeniu de studiu. Datorită acestui fapt, în momentul în care un domeniu este ștearsă vor fi șterse și informațiile tuturor seriilor ce aparțin domeniului respectiv prin intermediul unei constrângeri ON DELETE CASCADE asupra cheii externe “id_domeniu” din tabelul SERII.

Un domeniu nu poate exista fără să fie inclus într-o formă de învățământ. Datorită acestui fapt, în momentul în care o formă de învățământ este ștearsă vor fi șterse toate domeniile asociate formei de învățământ respective cu ajutorul unei constrângeri ON DELETE CASCADE asupra cheii externe “id_forma_invatamant” din tabelul DOMENII.

Nu este necesară păstrarea informațiilor despre notele unui student care a fost ștears. De aceea în tabelul NOTE există o constrângere de tipul ON DELETE CASCADE asupra cheii externe “id_student”.

Nu este necesară păstrarea informațiilor despre notele unui student la o materie care nu mai există. De aceea în tabelul NOTE există o constrângere de tipul ON DELETE CASCADE asupra cheii externe “id_materie”.

În momentul în care un angajat a fost șters, iar acel angajat predă o materie, nu trebuie ștearsă și materia respectivă, întrucât poate fi asociat un alt angajat capabil să finalizeze procesul educațional. De aceea în tabelul MATERII există o constrângere de tipul ON DELETE SET NULL asupra cheii externe “id_angajat”.

Angajații sunt grupați în departamente, în momentul ștergerii unui departament vor fi șterse și toate informațiile angajaților ce lucrează în departamentul respectiv. Acest lucru este implementat cu ajutorul unei constrângeri asupra cheii externe “id_domeniu” ON DELETE CASCADE din tabela ANGAJATI.

4. Creare tabelelor:

Tabelul Capacitate a fost creat folosind codul:

```
CREATE TABLE Capacitati (  
    tip_sala VARCHAR(40),  
    capacitate NUMBER(6,0) NOT NULL,  
    PRIMARY KEY (tip_sala),  
    CHECK(capacitate > 0),  
    CONSTRAINT uq_capacitati UNIQUE(tip_sala, capacitate)  
);
```

Tabelul Departamente a fost creat folosind codul:

```
CREATE TABLE Departamente (  
    id_departament NUMBER(6,0),  
    nume_departament VARCHAR(40) NOT NULL UNIQUE,  
    PRIMARY KEY (id_departament)  
);
```

Tabelul Posturi a fost creat folosind codul:

```
CREATE TABLE Posturi (  
    id_post NUMBER(6,0),  
    denumire VARCHAR(40) NOT NULL UNIQUE,  
    PRIMARY KEY (id_post)  
);
```


Tabelul Angajati a fost creat folosind codul:

```
CREATE TABLE Angajati (  
    id_angajat NUMBER(6,0),  
    nume VARCHAR(40) NOT NULL,  
    prenume VARCHAR(40) NOT NULL,  
    id_departament NUMBER(6,0) NOT NULL,  
    id_post NUMBER(6,0) NOT NULL,  
    salariu NUMBER(10,4) NOT NULL,  
    email VARCHAR(40) NOT NULL,  
    PRIMARY KEY (id_angajat),  
    CONSTRAINT fk_angajati_departament FOREIGN KEY(id_departament)  
        REFERENCES Departamente(id_departament)  
        ON DELETE CASCADE,  
    CONSTRAINT fk_angajati_posturi FOREIGN KEY(id_post)  
        REFERENCES Posturi(id_post),  
    CHECK(salariu > 0)  
);
```

Tabelul Materii a fost creat folosind codul:

```
CREATE TABLE Materii (  
    id_materie NUMBER(6,0),  
    nume_materie VARCHAR(40) NOT NULL UNIQUE,  
    nr_credite NUMBER(4,0) NOT NULL,  
    forma_examinare VARCHAR(40) NOT NULL,  
    id_angajat NUMBER(6,0),  
    PRIMARY KEY (id_materie),  
    CONSTRAINT fk_materii FOREIGN KEY(id_angajat) REFERENCES  
Angajati(id_angajat)  
);
```

Tabelul Sali a fost creat folosind codul:

```
CREATE TABLE Sali (  
    id_sala NUMBER(6,0),  
    tip_sala VARCHAR(40) NOT NULL,  
    id_departament NUMBER(6,0) NOT NULL,  
    PRIMARY KEY (id_sala),  
    CONSTRAINT fk_sali_dep FOREIGN KEY(id_departament) REFERENCES  
Departamente(id_departament),  
    CONSTRAINT fk_sali_tip FOREIGN KEY(tip_sala) REFERENCES  
Capacitati(tip_sala)  
);
```

Tabelul Note a fost creat folosind codul:

```
CREATE TABLE Note (  
    id_student NUMBER(6,0),  
    id_materie NUMBER(6,0),  
    data_examinare DATE,  
    nota NUMBER(10,2) NOT NULL,  
    CONSTRAINT cpk_note PRIMARY KEY (data_examinare, id_student,  
id_materie),  
    CONSTRAINT fk_note_student FOREIGN KEY(id_student) REFERENCES  
Studenti(id_student),  
    CONSTRAINT fk_note_materie FOREIGN KEY(id_materie) REFERENCES  
Materii(id_materie),  
    CHECK(nota BETWEEN 0 AND 10)  
);
```

Tabelul Forme_Invatamant a fost creat folosind codul:

```
CREATE TABLE Forme_Invatamant (  
    id_forma_invatamant VARCHAR(40),  
    denumire VARCHAR(40) NOT NULL UNIQUE,  
    PRIMARY KEY (id_forma_invatamant)  
);
```

Tabelul Domenii a fost creat folosind codul:

```
CREATE TABLE Domenii (  
    id_domeniu NUMBER(6,0),  
    denumire VARCHAR(40) NOT NULL,  
    numar_ani NUMBER(4,0) NOT NULL,  
    id_forma_invatamant VARCHAR(16) NOT NULL,  
    CONSTRAINT fk_domeniu_fi FOREIGN KEY(id_forma_invatamant)  
REFERENCES Forme_invatamant(id_forma_invatamant)  
ON DELETE CASCADE,  
    CHECK(numar_ani > 0),  
    PRIMARY KEY (id_domeniu)  
);
```

Tabelul Serii a fost creat folosind codul:

```
CREATE TABLE Serii (  
    id_serie NUMBER(6,0),  
    data_finalizare_studii DATE NOT NULL,  
    id_domeniu NUMBER(6,0) NOT NULL,  
    PRIMARY KEY (id_serie),  
    CONSTRAINT fk_serie_domeniu FOREIGN KEY(id_domeniu)  
REFERENCES Domenii(id_domeniu)  
ON DELETE CASCADE  
);
```

Tabelul Studenti a fost creat folosind codul:

```
CREATE TABLE Studenti (  
    id_student NUMBER(6,0),  
    nume VARCHAR(40) NOT NULL,  
    prenume VARCHAR(40) NOT NULL,  
    data_nastere DATE NOT NULL,  
    nationalitate VARCHAR(40) NOT NULL,  
    sex VARCHAR(1) NOT NULL,  
    email VARCHAR(40) NOT NULL UNIQUE,  
    id_serie NUMBER(6,0) NOT NULL,  
    taxa VARCHAR(2) NOT NULL,  
    PRIMARY KEY (id_student),  
    CHECK(sex IN ('M', 'F')),  
    CHECK(taxa IN ('DA', 'NU')),  
    CONSTRAINT fk_student_serie FOREIGN KEY(id_serie)  
        REFERENCES Serii(id_serie)  
        ON DELETE CASCADE  
);
```

5. Prezentarea modelului și a regulilor acestuia

Datele au fost inserate în tabele folosind următorul cod:

```
INSERT INTO Studenti  
VALUES(1, 'Grigoruta', 'Cosmin', to_date('15-04-1999', 'dd-mm-yyyy'),  
'romana', 'M', 'grigorutacosmin@universitate.ro', 100, 'NU');
```

```
INSERT INTO Serii  
VALUES(100, to_date('2021', 'yyyy'), 1);
```

```
INSERT INTO Domenii  
VALUES(1, 'Informatica', 3, 'IF');
```

```
INSERT INTO Forme_Invatamant  
VALUES('IF', 'Invatamant Frecventa');
```

```
INSERT INTO Note  
VALUES(1, 1, to_date('22-01-2020', 'dd-mm-yyyy'), 9.6);
```

```
INSERT INTO Sali  
VALUES(120, 'L', 1);
```

```
INSERT INTO Materii  
VALUES(1, 'Programare Procedurala', 5, 'EXAMEN', 1);
```

```
INSERT INTO Angajati  
VALUES(1, 'Rusu', 'Ioan', 1, 1, 5000.0, 'rusuioan@uni.ro');
```

```
INSERT INTO Posturi  
VALUES(1, 'Lector');
```

```
INSERT INTO Departamente  
VALUES(1, 'INFORMATICA');
```

```
INSERT INTO Capacitati  
VALUES('L', 25);
```