

## Proiect final

### 1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

Modelul real este Universitatea din Bucuresti, care este compusa din mai multe facultati.

Fiecare facultate are profesori, firme partenere, o asociatie studenteasca si specializari.

Fiecare specializare e compusa din serii si cursuri.

Fiecare serie e compusa din grupe, iar grupele sunt compuse din studenti.

Asocierea dintre profesori, serii si cursuri rezulta intr-o lectie de predare.

Din acest model putem deduce diverse lucruri ca: cati elevi are universitatea, cate cursuri sun predate in cadrul, cati profesori sunt angajati etc.

### 2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Fiecare facultate are asociata o unica asociatie studenteasca.

O facultate poate lucra cu mai multe firme, dar firma nu poate lucra cu mai multe facultati.

Fiecare facultate poate avea mai multi profesori, care lucreaza la mai multe facultati.

O facultate are mai multe specializari, dar acestea sunt diferite de la o facultate la alta.

Fiecare specializare are mai multe serii, care sunt diferite de la o specializare la alta.

Fiecare serie are mai multe grupe si fiecare grupa e formata din mai multi studenti.

O specializare are mai multe cursuri, iar un curs poate fi predat la mai multe specializari.

Fiecare curs poate fi predat de mai multi profesori la mai multe serii, fiecarei serii ii pot fi predate mai multe cursuri de mai multi profesori, iar fiecare profesor poate preda mai multe cursuri la mai multe serii.

### 3. Descrierea entităților, incluzând precizarea cheii primare.

Facultate- PK facultate\_id, contine diverse informatii despre facultate(ume, loc\_national)

Asociatii\_studentesti-PK Asociatie\_id, contine diverse informatie despre asociate(ume, reprezentant)

Firme\_partenere-PK Firma\_id, contine diverse informatii despre firma(ume, grad de intelegere)

Profesori-PK Profesor\_id, contine diverse informatii despre profesor(ume, email)

Specializari-PK Specializare\_id, contine diverse informatii despre specializare(ume, durata in ani de studiu)

Cursuri-PK Curs\_id, contine diverse informatii despre cursuri(rata\_promovabilitate, daca e cu pauza sau nu)

Serii-PK Serie\_id, contine diverse informatii despre serie(numar serie, sef serie)

Grupe-PK Grupa\_id, contine diverse informatii despre grupa(numar grupa, sef grupa)

Studenti- PK Student\_id, contine diverse informatii despre student(media, numele)

### 4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Facultate 1 to 1 Asociatii\_studentesti

Facultate 1 to many Firme\_partenere

Facultate many to many Profesori

Facultate 1 to many Specializari

Specializari many to many Cursuri

Specializari 1 to many Serii

Serii 1 to many Grupe

Grupe 1 to many Studenti

(Profesori, Cursuri, Serii) many to many to many

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

Firme\_partenere:

- firma\_id intreg
- grad de intelegere procent, permite 2 zecimale dupa virgula
- nume\_firma sir de caractere
- facultate\_id intreg

Asociatii\_studentesti:

- asociatie\_id intreg
- nume\_reprezentant sir de caractere
- nume\_asociatie sir de caractere
- facultate\_id intreg

Facultate:

- facultate\_id intreg
- nume\_facultate sir de caractere
- adresa sir de caractere
- loc national interg

Profesori:

- profesor\_id intreg
- facultate\_id intreg
- nume\_profesor, prenume\_profesor siruri de caractere
- data\_angajare e de tip date
- email\_profesor sir de caractere

Specializari:

- specializare\_id intreg
- durata intreg
- nspecializare\_nume sir de caractere
- facultate\_id intreg
- probabilitate\_angajare procent, permite 2 zecimale dupa virgula

Cursuri:

- curs\_id intreg
- rata\_promovabilitate procent, permite 2 zecimale dupa virgula
- pauza sir de caractere: valori posibile: DA/NU

Serii:

- serie\_id intreg
- numar\_serie intreg
- nume\_sef\_serie, email\_serie sir de caractere
- specializare\_id intreg

Grupe:

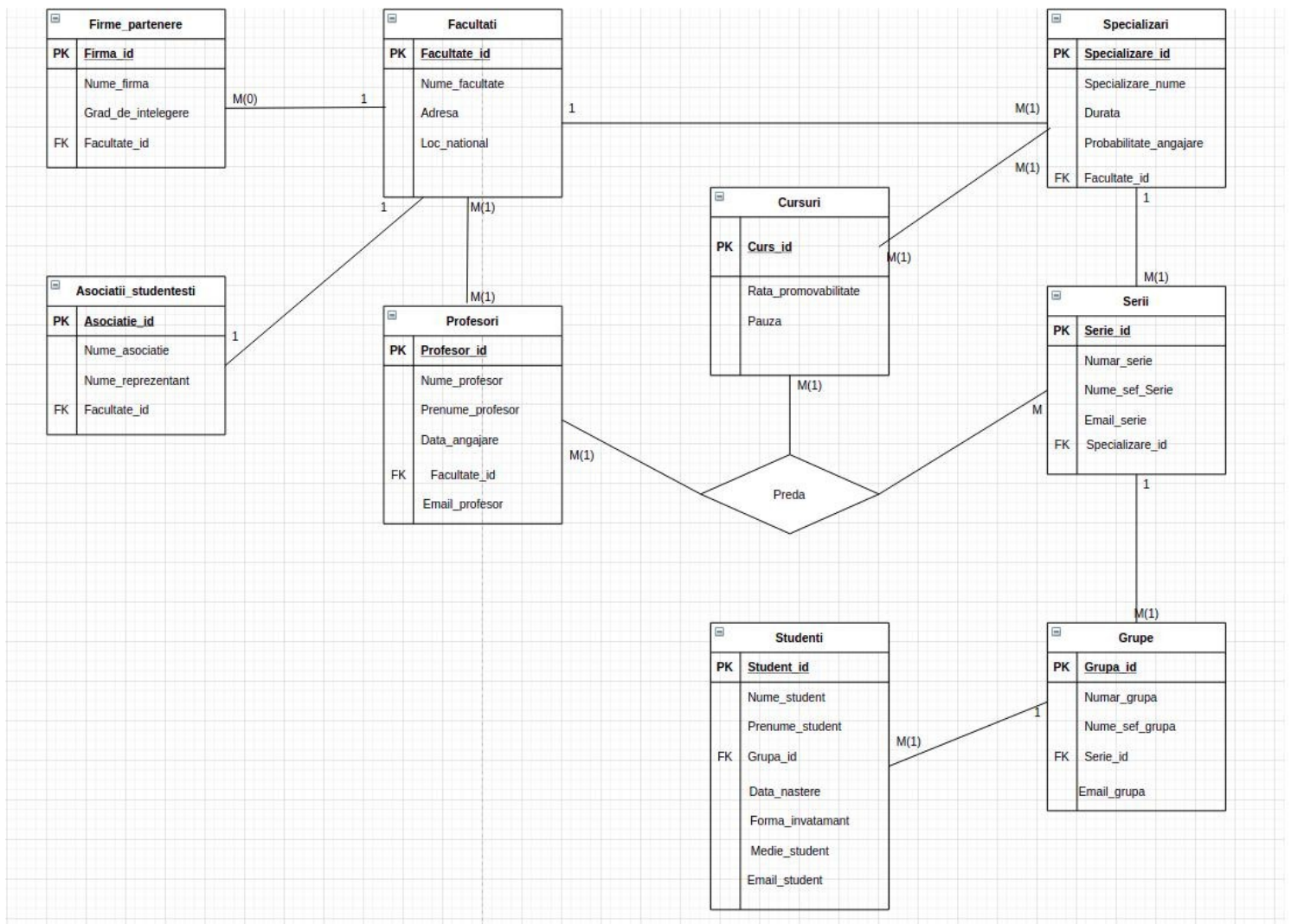
- grupa\_id intreg
- numar\_grupa intreg

-nume\_sef\_grupa,email\_grupa sir de caractere  
 -serie\_id intreg

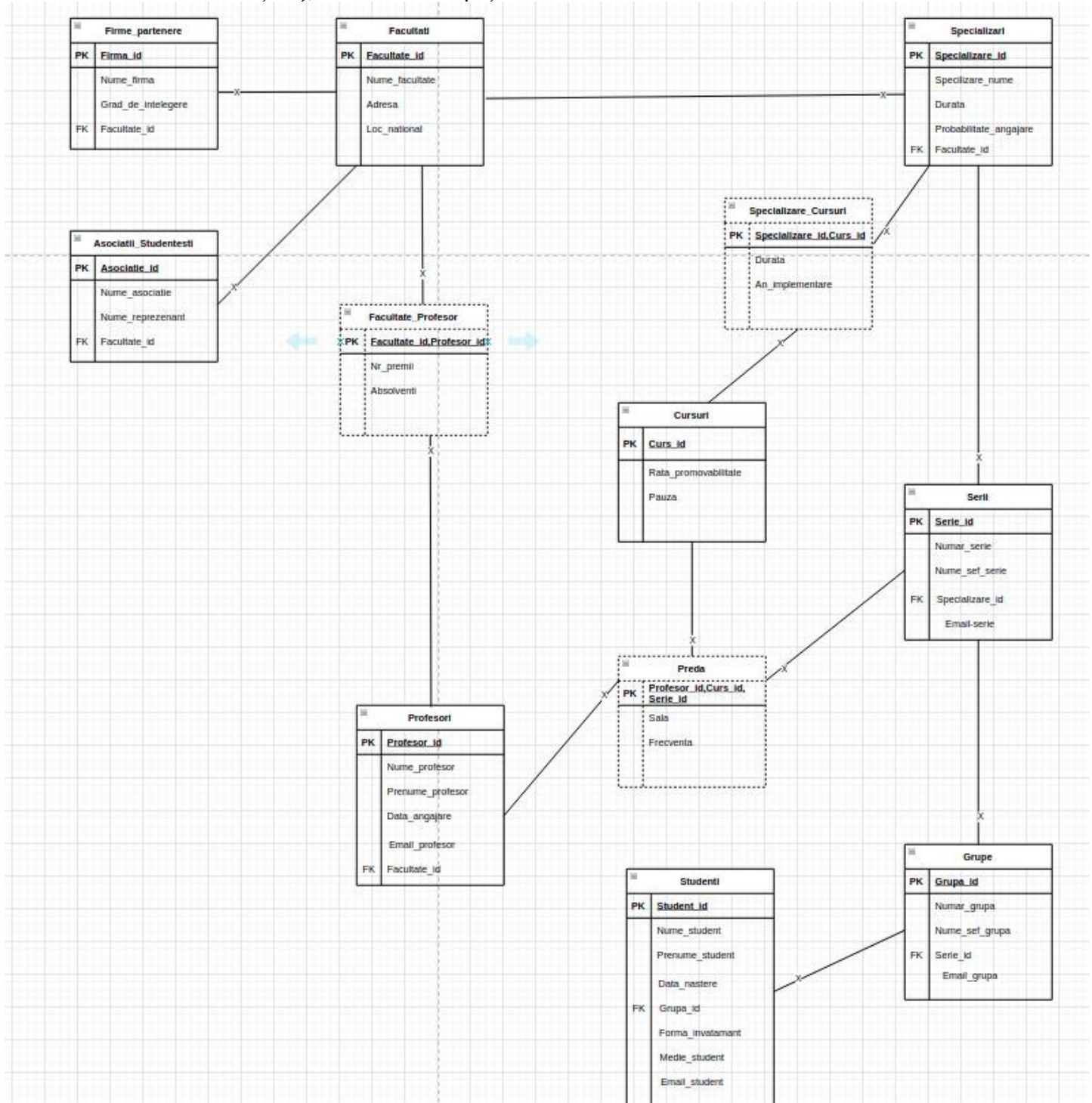
Studenti:

-student\_id intreg,  
 -nume\_student, prenume\_student, forma\_invatamant,email\_student siruri de caractere  
 -grupa\_id intreg  
 -data\_nastere de tip date  
 -medie\_student e mai mica sau egala decat 10, permite 2 zecimale dupa virgula

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

Facultati(PK Firma\_id,, Nume\_facultate, adresa, Loc\_national)  
Firme\_partenere(PK Firma\_id, Nume\_firma, Grad\_de\_intelegere, FK Facultate\_id)  
Asociatii\_studentesti(PK Asociatie\_id, FK Facultate\_id, Nume\_asociatie, Nume\_reprezentant)  
Facultate\_profesor(PK Facultate\_id, PK Profesor\_id, Nr\_premii, Absolventi)  
Profesori(PK Profesor\_id, FK Facultate\_id, Nume\_profesor, Prenume\_profesor, Data\_angajare, Email\_profesor)  
Specializari(PK Specializare\_id, FK Facultate\_id, Specializare\_nume, Durata, Probabilitate\_angajare)  
Cursuri(PK Curs\_id, Rata\_promovabilitate, pauza)  
Specializari\_cursuri(PK Specializare\_id, PK Curs\_id, Durata, An\_implementare)  
Serii(PK Serie\_id, FK Specializare\_id, Numar\_serie, Nume\_sef\_Serie, Email\_serie)  
Grupe(PK Grupa\_id, FK Serie\_id, Numar\_grupa, Nume\_sef\_grupa, Email\_grupa)  
Studenti(PK Student\_id, FK Grupa\_id, Nume\_student, Prenume\_student, Data\_nastere, Forma\_invatamant, Medie\_student, Email\_studenti)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

NON-FN1:

Adaugam atributul Adresa in tabelul Studenti si acesta contine, adresa domiciliului si adresa din Bucuresti a studentului.

FN1:

Rupem acest atribut Adresa in attributele Adresa\_Bucuresti, Adresa\_domiciliu.

NON-FN2:

Specializare\_cursuri(PK Specializare\_id, PK Curs\_id, An\_implementare, Nume\_curs)

K1=Specializare\_id

K2=Curs\_id

X=An\_implementare

Y=Nume\_curs(depinde direct de K2)

FN2:

Specializare\_cursuri(PK Specializare\_id, PK Curs\_id, An\_implementare)

Cursuri(PK Curs\_id, Nume\_curs)

NON-FN3:

Specializari(PK Specializare\_id, Facultate\_id, Durata, Nume\_facultate)

K=Specializare\_id

X=Facultate\_id

Y=Durata

Z=Nume\_facultate

Z depinde de X

X,Y depind de K

FN3:

Specializari(PK Specializare\_id, Facultate\_id, Durata)

Facultate(PK Facultate\_id, Nume\_facultate)

18. a. Realizarea normalizării BCNF, FN4, FN5.

NON-BCNF:

Specializare\_Curs(PK Specializare\_id, PK Curs\_id, An\_studiu, Modul\_Curs)

K1=Specializare\_id

K2=Curs\_id

X=Modul\_curs

Y=An\_studiu

Y depinde de (K1,K2)

K2 depinde de X

BCNF:

Specializare\_Curs(PK Specializare\_id, PK Curs\_id, An studiu)

Curs(PK Modul\_curs, Curs\_id)

NON-FN4:

Relatii\_externe(Student\_id, Limba\_straina, Email)

Student_id	Limba_straina	Email
10	y	a
10	y	b
10	z	b

$10 \rightarrow y \rightarrow b(1,1,2)$

$10 \rightarrow y \rightarrow b(1,2,3)$

FN4:

Student1(Student\_id, Limba\_straina)

Student2(Student\_id, Email)

NON-FN5:

Preda(PK Curs\_id, PK Profesor\_id, PK Serie\_id)

Daca cele 3 chei n-ar fi unice

Curs_id	Profesor_id	Serie_id
10	15	100
10	20	101

Curs_id	Profesor_id
10	15
10	20

Profesor_id	Serie_id
15	100
20	101

Curs_id	Serie_id
10	100
10	101

Din tabelele Curs\_profesor si Curs\_serie reiese linia 10 15 101 in plus

FN5:

Preda(PK Curs\_id, PK Profesor\_id, PK Serie\_id), cand cheile sunt unice nu mai avem stari in plus.