

UNIVERSITATEA DIN BUCUREȘTI  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**PROIECT BAZE DE DATE**

PROFESOR COORDONATOR:

VASILE SILVIU-LAURENȚIU

STUDENT:

HOLMANU ANTONIO-MARIUS

BUCUREȘTI

2022

UNIVERSITATEA DIN BUCUREȘTI  
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ  
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**SISTEM DE VOT ELECTRONIC**

PROFESOR COORDONATOR:

VASILE SILVIU-LAURENȚIU

STUDENT:

HOLMANU ANTONIO-MARIUS

BUCUREȘTI

2022

## CUPRINS

1. Prezentarea modelului.....	1
2. Regulile modelului.....	3
3. Diagramă Entitate-Relație.....	6
3.1. Reprezentare.....	6
3.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților.....	7
3.2.1. Descrierea entităților, atributelor și a cheilor .....	7
3.2.2. Descrierea relațiilor și a cardinalităților .....	10
4. Diagramă conceptuală.....	14
4.1. Reprezentare.....	14
4.2. Descrierea constrângerilor de integritate.....	14
4.2.1. Tabelul FIRMA .....	14
4.2.2. Tabelul PROIECT .....	15
4.2.3. Tabelul SARCINA .....	15
4.2.4. Tabelul DEPARTAMENT .....	16
4.2.5. Tabelul ECHIPA.....	16
4.2.6. Tabelul STUDENT .....	17
4.2.7. Tabelul STUDENT_ECHIPA (tabel asociativ STUDENT-ECHIPA).....	17
4.2.8. Tabelul CANDIDATURA (tabel asociativ ECHIPA-SARCINA).....	17
4.2.9. Tabelul SCRUTIN .....	18
4.2.10. Tabelul VOT (tabel asociativ STUDENT-SCRUTIN) .....	19
4.3. Schemele relaționale .....	20
4.4. Descrierea constrângerilor ON DELETE CASCADE / SET NULL .....	20
5. Script.....	22
5.1. Prima etapă: <b>DROP SEQUENCIES, ALTER TABLES</b> și <b>DROP TABLES</b> .....	22
5.2. Etapa a doua: Crearea secvențelor, a tabelelor și a constrângerilor .....	23
5.2.1. Tabelul <b>FIRMA</b> .....	23
5.2.2. Tabelul <b>PROIECT</b> .....	24
5.2.3. Tabelul <b>SARCINA</b> .....	25
5.2.4. Tabelul <b>DEPARTAMENT</b> .....	26
5.2.5. Tabelul <b>ECHIPA</b> .....	26
5.2.6. Tabelul <b>STUDENT</b> .....	27
5.2.7. Tabelul asociativ <b>STUDENT_ECHIPA</b> .....	28
5.2.8. Tabelul asociativ <b>CANDIDATURA</b> .....	28

5.2.9. Tabelul <b>SCRUTIN</b> .....	29
5.2.10. Tabelul asociativ <b>VOT</b> .....	30
5.3. Ultima etapa: Introducerea datelor .....	31
5.3.1. Tabelul <b>FIRMA</b> .....	31
5.3.2. Tabelul <b>PROIECTE</b> .....	32
5.3.3. Tabelul <b>SARCINA</b> .....	32
5.3.4. Tabelul <b>DEPARTAMENT</b> .....	33
5.3.5. Tabelul <b>STUDENT</b> .....	34
5.3.6. Tabelul <b>ECHIPA</b> .....	34
5.3.7. Tabelul asociativ <b>STUDENT_ECHIPA</b> .....	34
5.3.8. Tabelul asociativ <b>CANDIDATURA</b> .....	35
5.3.9. Tabelul <b>SCRUTIN</b> .....	35
5.3.10. Tabelul asociativ <b>VOT</b> .....	36

## 1. Prezentarea modelului

Asociația Studenților la Matematică și Informatică (ASMI) și anumite firme cu care facultatea se află în parteneriat și-au propus să aducă mai aproape de studenți modul de lucru al angajaților din domeniul IT. Ei vor să le ofere studenților șansa de a căpăta experiența organizării și a lucrului în echipă atunci când vine vorba despre un proiect mai complex (cu un anumit buget) pe care îl au de realizat. Astfel, firmele sunt invitate să propună proiecte și sarcini de îndeplinit oricând doresc, însă, pentru a realiza o selecție a membrilor asociației care vor lucra la aceste proiecte, se vor supune la vot echipe de studenți (formate din membri ai oricărui departament al asociației) care vor fi responsabile pentru sarcini din diverse proiecte. Studenții vor trebui să folosească un sistem electronic de vot pentru a-și exprima preferințele.

Câteva precizări înaintea definirii regulilor modelului propus:

- ❖ În ceea ce privește partea de candidatură a studenților în echipe:

Se consideră că echipa este forma de organizare necesară pentru a putea ca un student să candideze la o sarcină. Inițial se definesc echipele, iar ulterior, studenții se pot înscrie în echipele respective.

- Justificare: Pentru anumite sarcini pot fi înscrise în procesul de alegere un număr maxim de echipe. Astfel, pentru a oferi oportunitatea unui student de a lucra la o anumită sarcină, se vor defini mai întâi echipele.

- ❖ În ceea ce privește înscrierea candidaturii unei echipe la o sarcină:

În momentul în care o echipă dorește să se înscrie în cursa electorală pentru obținerea unei sarcini, acelei echipe i se va atribui o poziție pe buletinul de vot. Această poziție este chiar numărul de ordine al înscrierii echipei pe listele electorale pentru sarcina respectivă.

De asemenea, validarea înscrierii candidaturii unei echipe se face doar dacă data înscrierii în competiție este cuprinsă între data începerii înscrierilor și data finalizării înscrierilor (inclusiv). Prin validare se înțelege luarea în calcul a voturilor obținute de o echipă dacă îndeplinește condiția anterior enunțată.

- ❖ În ceea ce privește partea de vot:

Pentru fiecare sarcină dintr-un anumit proiect va exista câte un scrutin.

(Excepție: Pot exista sarcini la se înscrie doar o echipă în perioada de depunere a candidaturilor; astfel echipa respectiva va fi considerată câștigătoare și nu mai este nevoie să se mai inițieze un scrutin. Totuși, chiar dacă echipa este declarată câștigătoare, este posibil să se ceară un sondaj orientativ (oferit prin vot) cu numărul de studenți care susțin acea echipă pentru sarcina respectivă. Deci, sunt situații în care se poate iniția un scrutin sau nu pentru o sarcină.)

Fiecare scrutin va avea câte un token. Un student se va putea autentifica în vederea votului la un scrutin cu numărul său matricol și cu token-ul caracteristic scrutinului. De asemenea, se consideră că în urma unei campanii electorale desfășurate de echipele care candidează pentru o sarcină, fiecare alegător cunoaște de dinaintea începerii procesului de vot poziția caracteristică pe buletinul de vot a echipei pe care dorește să o voteze.

Ideea modelului propus implică ca un student să voteze în cadrul unui scrutin, o singură echipă pe baza poziției candidaturii echipei preferate.

## 2. Regulile modelului

1. Un student trebuie să facă parte dintr-un singur departament.
2. Un departament poate să aibe sau nu studenți înmatriculați.  
(Justificare: Pot exista departamente care încă nu și-au recrutat studenți.)
3. O echipă poate să aibă sau nu studenți.  
(Justificare: Sunt situații în care prima dată se definește echipa, și ulterior, studenții se înscriu în acea echipă.)
4. Un student poate să fie inclus sau nu într-o echipă.  
(Justificare: Studentul dorește să rămână un simplu alegător în procesul de vot, adică nu vrea să se implice în partea de candidaturi pentru anumite sarcini.)
5. Un student poate să fie inclus în mai multe echipe.
6. Un proiect trebuie să aparțină unei singure firme.
7. O firmă poate propune sau nu proiecte.  
(Justificare: Deși firma se află în parteneriat cu facultatea, există posibilitatea să nu dorească să participe la această inițiativă de propunere a unor proiecte și sarcini pentru studenți.)
8. O sarcină trebuie să aparțină unui singur proiect.
9. Un proiect poate să aibe sau nu sarcini de îndeplinit.  
(Justificare: Deși firma a inițiat un proiect pentru studenți, există posibilitatea ca bugetul pe care l-a anunțat să nu fi fost respectat sau să poată acoperi finanțarea unor sarcini elementare din proiectul propus.)
10. O echipă poate concura sau nu pentru sarcini din orice proiect al oricărei firme.  
(Justificare: Există posibilitatea ca o echipă să nu dorească să se înscrie la nicio sarcină dintr-un anumit proiect din motive proprii.)
11. Pentru o sarcină pot concura sau nu echipe.  
(Justificare: O firmă poate propune anumite cerințe pentru a putea candida la o sarcină, dar din anumite motive, nicio echipă nu îndeplinește condițiile necesare.)
12. Un scrutin trebuie să fie inițiat pentru o singură sarcină.
13. O sarcină poate să implice sau nu scrutine.

*(Justificare: Dacă pentru un scrutin nu s-a întrunit o majoritate a voturilor pentru o opțiune, atunci va exista un nou scrutin pentru aceeași sarcină cu anumite echipe din primul scrutin.*

*Justificare: Există situații în care pentru o sarcină candidează o echipă, iar scrutinul poate să nu mai aibă loc, întrucât echipa respectivă este declarată câștigătoare.)*

14. În funcție de ordinea înscrierii candidaturii pentru o anumită sarcină (într-o anumită perioadă de înscrieri), echipelor li se atribuie o poziție la alegeri.

*(Exemplu: Echipa „Vulturii Albaștri” care s-a înscris a 5-a pe lista de candidați pentru sarcina „Programare în Go” dintr-un anumit proiect, va avea poziția 5 în cadrul scrutinului pentru desemnarea unei echipe câștigătoare la acea sarcină.*

*Justificare: Poziția unei echipe calculată în acest mod oferă un argument obiectiv pentru modalitatea de atribuire a pozițiilor echipelor în cadrul votului. O echipă își va înscrie candidatura la o sarcină cu atât mai repede, cu cât consideră că este cea mai potrivită pentru acea sarcină.)*

15. Un student care dorește să-și exercite dreptul de vot, va putea vota la un scrutin pe baza unui identificator unic al scrutinului la care va avea acces.

*(Exemplu: token\_scrutin reprezintă parola pentru care studentul își poate vota echipa favorită la un anumit scrutin.*

*Justificare: Există posibilitatea să se desfășoare mai multe scrutine în același timp.)*

16. Se consideră că un student votează o singură echipă (mai exact, poziția candidaturii echipei) pentru o sarcină.

*(Justificare: Se consideră că fiecare alegător cunoaște de dinaintea începerii votului, poziția candidaturii echipei pe care dorește să o voteze.)*

17. Un student poate vota sau nu la scrutine.

*(Justificare: Se consideră că dacă un student pune o altă poziție decât cele existente ale echipelor într-un scrutin, opțiunea sa de vot este considerată invalidă, adică nu va fi numărată la finalul procesului de vot.)*

18. La un scrutin pot vota sau nu studenți.

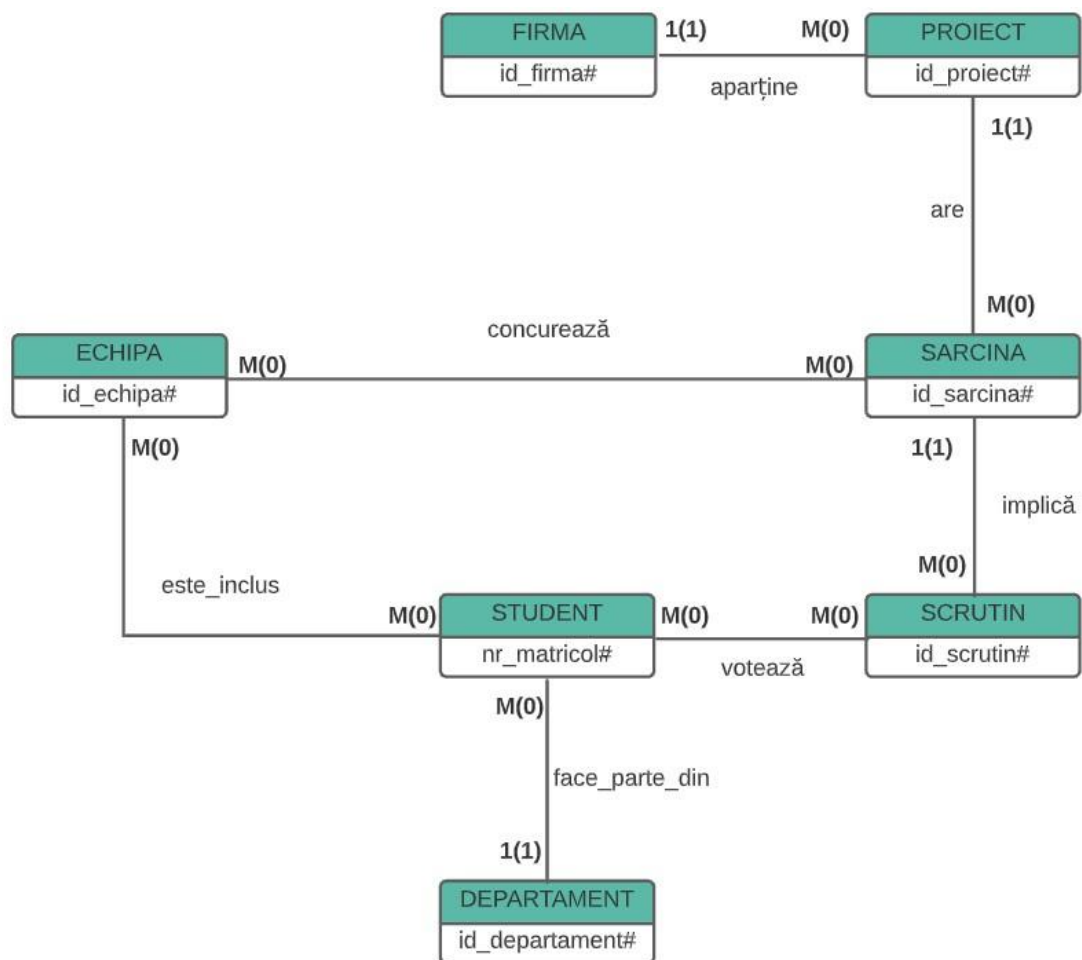
*(Justificare: Cum studenții cunosc de dinaintea începerii procesului de vot pozițiile echipelor care candidează pentru sarcini, este posibil ca pentru o*



*sarcină să nu candideze nicio echipă. Astfel, niciun student nu ar vota la un scrutin pentru care nu candidează nimeni.)*

### 3. Diagramă Entitate-Relație

#### 3.1. Reprezentare



### 3.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

#### 3.2.1. Descrierea entităților, atributelor și a cheilor

##### 3.2.1.1. Entitatea FIRMA

Entitatea FIRMA stochează datele firmelor partenere cu facultatea și care propun și finanțează anumite proiecte.

Cheie	Atribut	Descriere	Tip de dată
PK	id_firma	Identificator unic	NUMBER(3,0)
	nume_firma	Numele firmei	VARCHAR2 (100 CHAR)
	email	E-mailul firmei pentru informații suplimentare legate de proiectele propuse	VARCHAR2 (40 CHAR)
	telefon	Numărul de telefon la care pot fi contactați reprezentanții ai firmei. Se presupune că sunt numere de România (10 cifre)	VARCHAR2 (10 CHAR)

##### 3.2.1.2. Entitatea PROIECT

Entitatea PROIECT stochează detaliile unui proiect propus de o firmă.

Cheie	Atribut	Descriere	Tip de dată
PK	id_proiect	Identificator unic	NUMBER(4,0)
FK	id_firma	Firma care propune proiectul	NUMBER(3,0)
	nume_proiect	Numele proiectului	VARCHAR2 (80 CHAR)
	buget	Suma de bani alocată de firmă pentru implementarea proiectului	NUMBER(4,0)

#### 3.2.1.3. Entitatea SARCINA

Entitatea SARCINA caracterizează sarcinile componente ale proiectului.

Cheie	Atribut	Descriere	Tip de dată
PK	id_sarcina	Identificator unic	NUMBER(5,0)
FK	id_proiect	Proiectul din care provine sarcina	NUMBER(4,0)
	nume_sarcina	Numele sarcinii	VARCHAR2 (100 CHAR)

#### 3.2.1.4. Entitatea DEPARTAMENT

Entitatea DEPARTAMENT stochează departamentele din asociație.

Cheie	Atribut	Descriere	Tip de dată
PK	id_departament	Identificator unic	NUMBER(3,0)
	nume_departament	Numele departamentului	VARCHAR2 (30 CHAR)

#### 3.2.1.5. Entitatea ECHIPA

Entitatea ECHIPA memorează datele despre o echipă înscrisă în competiția pentru obținerea uneia sau a mai multor sarcini din cadrul unui proiect.

Cheie	Atribut	Descriere	Tip de dată
PK	id echipa	Identificator unic	NUMBER(4,0)
	nume echipa	Numele echipei	VARCHAR2 (60 CHAR)

#### 3.2.1.6. Entitatea STUDENT

Entitatea STUDENT stochează informațiile despre un student (membru al asociației) cu drept de vot.

Cheie	Atribut	Descriere	Tip de dată
PK	nr_matricol	Identificator unic format din 5 cifre	NUMBER(5,0)

FK	id_departament	Departamentul de care aparține studentul	NUMBER(3,0)
	nume	Numele studentului	VARCHAR2 (40 CHAR)
	prenume	Prenumele studentului	VARCHAR2 (50 CHAR)

### 3.2.1.7. Entitatea SCRUTIN

Entitatea SCRUTIN reține informațiile necesare părții de organizare a votului (perioadă de înscriere, data la care au loc alegerile) pentru o sarcină. De asemenea, reprezintă entitatea necesară pentru care un student își poate exprima opțiunea de vot. (prin intermediul unei parole unice)

Cheie	Atribut	Descriere	Tip de dată
PK	id_scrutin	Identificator unic	NUMBER(6,0)
FK	id_sarcina	Identificatorul sarcinii la care face referire	NUMBER(5,0)
	data_inceput_inscriere	Data la care încep înscrierile pentru candidatura la sarcină	DATE
	data_finalizare_inscriere	Data la care se termină înscrierile pentru candidatura la sarcină	DATE
	data_scrutin	Data la care are loc scrutinul. Se presupune că scrutinul începe și se termină în aceeași zi.	DATE
	token_scrutin	Parola pentru a putea vota în cadrul scrutinului	NUMBER(7,0)

### 3.2.2. Descrierea relațiilor și a cardinalităților

#### 3.2.2.1. *STUDENT – DEPARTAMENT*

Cardinalitate minimă

➔ Din câte departamente trebuie să facă parte un student?

1

➔ Câți studenți trebuie să facă parte dintr-un departament?

0

Cardinalitate maximă

➔ Din câte departamente poate să facă parte un student?

1

➔ Câți studenți pot să facă parte dintr-un departament?

MULȚI

#### 3.2.2.2. *STUDENT – ECHIPA*

*Observație:* Fiind o relație de M:N se va crea un tabel asociativ în diagrama conceptuală, numit *STUDENT\_ECHIPA*.

Cardinalitate minimă

➔ Câți studenți trebuie să fie incluși într-o echipă?

0

➔ În câte echipe trebuie să fie inclus un student?

0

Cardinalitate maximă

➔ Câți studenți pot să fie incluși într-o echipă?

MULȚI

➔ În câte echipe poate să fie inclus un student?

MULTE

### 3.2.2.3. FIRMA – PROIECT

Cardinalitate minimă

➔ De câte firme trebuie să aparțină un proiect?

1

➔ Câte proiecte trebuie să aibă o firmă?

0

Cardinalitate maximă

➔ De câte firme poate să aparțină un proiect?

1

➔ Câte proiecte poate să aibă o firmă?

MULTE

### 3.2.2.4. PROIECT – SARCINA

Cardinalitate minimă

➔ De câte proiecte trebuie să aparțină o sarcină?

1

➔ Câte sarcini trebuie să aibă un proiect?

0

Cardinalitate maximă

➔ De câte proiecte poate să aparțină o sarcină?

1

➔ Câte sarcini poate să aibă un proiect?

MULTE

### 3.2.2.5. ECHIPA – SARCINA

*Observație:* Fiind o relație de M:N se va crea un tabel asociativ în diagrama conceptuală, numit CANDIDATURA.

Cardinalitate minimă

➔ Câte echipe trebuie să concureze pentru o sarcină?

0

➔ La câte sarcini trebuie să concureze o echipă?

0

Cardinalitate maximă

➔ Câte echipe pot să concureze pentru o sarcină?

MULTE

➔ La câte sarcini poate să concureze o echipă?

MULTE

#### 3.2.2.6. SARCINA – SCRUTIN

Cardinalitate minimă

➔ Câte sarcini trebuie să fie implicate într-un scrutin?

1

➔ În câte scrutine trebuie să fie implicată o sarcină?

0

Cardinalitate maximă

➔ Câte sarcini pot să fie implicate într-un scrutin?

1

➔ În câte scrutine poate să fie implicată o sarcină?

MULTE

#### 3.2.2.7. STUDENT – SCRUTIN

*Observație:* Fiind o relație de M:N se va crea un tabel asociativ în diagrama conceptuală, numit VOT.

Cardinalitate minimă

➔ Câți studenți trebuie să voteze în cadrul unui scrutin?

0

➔ La câte scrutine trebuie să voteze un student?

0



Cardinalitate maximă

➔ Câți studenți pot să voteze în cadrul unui scrutin?

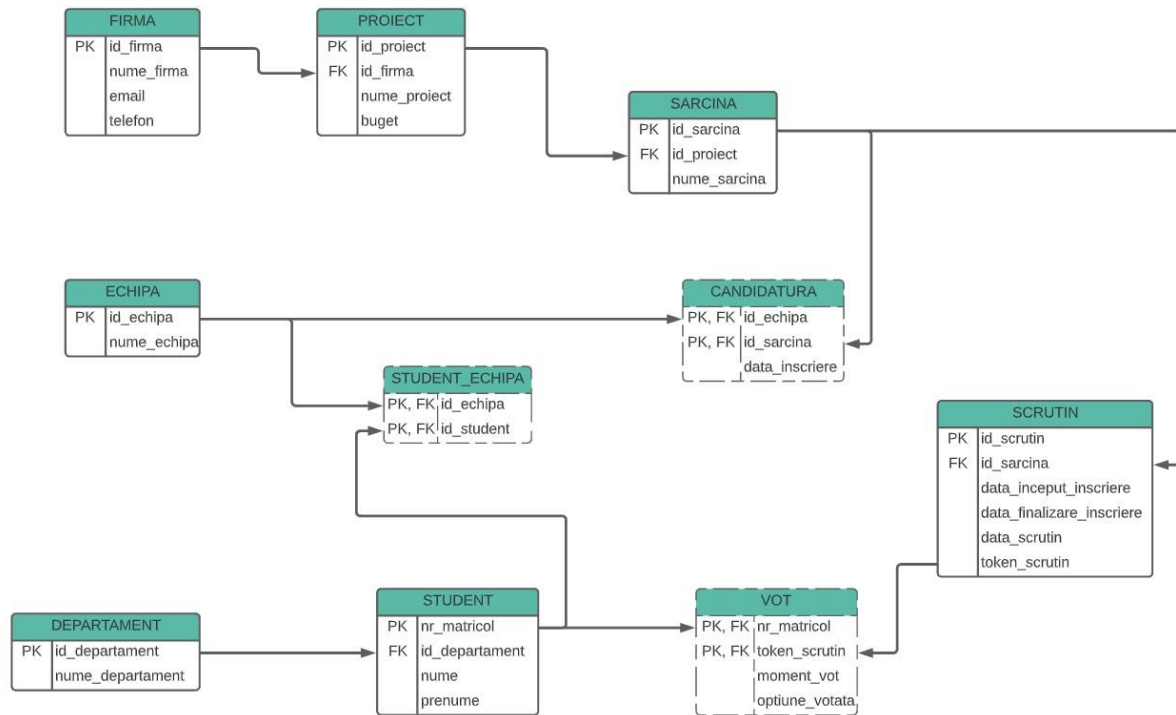
MULȚI

➔ La câte scrutine poate să voteze un student?

MULTE

## 4. Diagramă conceptuală

### 4.1. Reprezentare



### 4.2. Descrierea constrângerilor de integritate

#### 4.2.1. Tabelul FIRMA

##### ❖ Constrângeri de tip **PRIMARY KEY**

- FIRMA\_id\_firma\_PK – atribut „id\_firma”

##### ❖ Constrângeri de tip **NOT NULL**

- FIRMA\_nume\_firma\_NN - atribut „nume\_firma”
  - Firma trebuie să aibă un nume în mod obligatoriu
- FIRMA\_email\_NN – atribut „email”
  - Adresa de e-mail la care poți contacta firma este obligatorie
- FIRMA\_telefon\_NN - atribut „telefon”
  - Telefonul la care se poate intra în legătură directă cu firma este obligatoriu

##### ❖ Constrângeri de tip **UNIQUE**

- FIRMA\_num\_firma\_UK - atribut „nume\_firma”
  - Numele firmelor trebuie să fie unice
- FIRMA\_email\_UK – atribut „email”
  - Nu exista posibilitatea ca mai multe firme să poată fi contactate la aceeași adresă de e-mail
- FIRMA\_telefon\_UK - atribut „telefon”
  - Nu exista posibilitatea ca mai multe firme să poată fi contactate la același număr de telefon
- ❖ Constângeri de tip **CHECK**
  - FIRMA\_telefon\_CH - atribut „telefon”
    - Verific dacă numărul de telefon are exact 10 cifre (se presupune că numerele de telefon alocate de către firme sunt de România)
  - FIRMA\_email\_CH – atribut „email”
    - Verific dacă e-mailul este valid

#### 4.2.2. Tabelul PROIECT

- ❖ Constângeri de tip **PRIMARY KEY**
  - PROIECT\_id\_proiect\_PK – atribut „id\_proiect”
- ❖ Constângeri de tip **FOREIGN KEY**
  - PROIECT\_id\_firma\_FK – atribut „id\_firma”
    - Referențiază o coloană din tabelul FIRMA (id\_firma)
- ❖ Constângeri de tip **NOT NULL**
  - PROIECT\_id\_firma\_NN - atribut „id\_firma”
    - Proiectul trebuie să aparțină unei firme
  - PROIECT\_num\_proiect\_NN - atribut „nume\_proiect”
    - Proiectul trebuie să aibă un nume în mod obligatoriu
  - PROIECT\_buget\_NN - atribut „buget”
    - Bugetul alocat pentru un proiect este obligatoriu

#### 4.2.3. Tabelul SARCINA

- ❖ Constângeri de tip **PRIMARY KEY**
  - SARCINA\_id\_sarcina\_PK – atribut „id\_sarcina”
- ❖ Constângeri de tip **FOREIGN KEY**
  - SARCINA\_id\_proiect\_FK – atribut „id\_proiect”
    - Referențiază o coloană din tabelul PROIECT (id\_proiect)
- ❖ Constângeri de tip **NOT NULL**
  - SARCINA\_id\_proiect\_NN - atribut „id\_proiect”
    - Sarcina trebuie să aparțină unui proiect
  - SARCINA\_ume\_sarcina\_NN - atribut „ume\_sarcina”
    - Sarcina trebuie să aibă un ume în mod obligatoriu

#### 4.2.4. Tabelul DEPARTAMENT

- ❖ Constângeri de tip **PRIMARY KEY**
  - DEPARTAMENT\_id\_dep\_PK – atribut „id\_departament”
- ❖ Constângeri de tip **NOT NULL**
  - DEPARTAMENT\_ume\_dep\_NN - atribut „ume\_departament”
    - Departamentul trebuie să aibă un ume în mod obligatoriu
- ❖ Constângeri de tip **UNIQUE**
  - DEPARTAMENT\_ume\_dep\_UK - atribut „ume\_departament”
    - Numele departamentelor trebuie să fie unice

#### 4.2.5. Tabelul ECHIPA

- ❖ Constângeri de tip **PRIMARY KEY**
  - ECHIPA\_id echipa\_PK – atribut „id echipa”
- ❖ Constângeri de tip **NOT NULL**
  - ECHIPA\_ume echipa\_NN - atribut „ume echipa”
    - Echipa trebuie să aibă un ume în mod obligatoriu
- ❖ Constângeri de tip **UNIQUE**
  - ECHIPA\_ume echipa\_UK - atribut „ume echipa”
    - Numele echipelor trebuie să fie unice (pentru diversitatea numelor)

#### 4.2.6. Tabelul STUDENT

- ❖ Constângeri de tip **PRIMARY KEY**
  - STUDENT\_nr\_matricol\_PK – atribut „nr\_matricol”
- ❖ Constângeri de tip **FOREIGN KEY**
  - STUDENT\_id\_departament\_FK – atribut „id\_departament”
    - Referențiază o coloană din tabelul DEPARTAMENT (id\_departament)
- ❖ Constângeri de tip **NOT NULL**
  - STUDENT\_id\_departament\_NN - atribut „id\_departament”
    - Studentul trebuie să facă parte dintr-un departament
  - STUDENT\_ume\_NN - atribut „ume”
    - Studentul trebuie să aibă un ume în mod obligatoriu
  - STUDENT\_prenume\_NN – atribut „prenume”
    - Studentul trebuie să aibă un prenume în mod obligatoriu

#### 4.2.7. Tabelul STUDENT\_ECHIPA (tabel asociativ STUDENT-ECHIPA)

- ❖ Constângeri de tip **PRIMARY KEY**
  - STUDENT\_ECHIPA\_PK – attribute: „id echipa”, „id student” (cheie primară compusă)
- ❖ Constângeri de tip **FOREIGN KEY**
  - STUDENT\_id echipa\_FK – atribut „id echipa”
    - Referențiază o coloană din tabelul ECHIPA (id echipa)
  - STUDENT\_id student\_FK – atribut „id student”
    - Referențiază o coloană din tabelul STUDENT (nr\_matricol)

#### 4.2.8. Tabelul CANDIDATURA (tabel asociativ ECHIPA-SARCINA)

- ❖ Constângeri de tip **PRIMARY KEY**
  - CANDIDATURA\_PK – attribute: „id echipa” și „id sarcina” (cheie primară compusă)

❖ Constângeri de tip **FOREIGN KEY**

- CANDIDATURA\_id echipa\_FK – atribut „id echipa”
  - Referențiază o coloană din tabelul ECHIPA (id echipa)
- CANDIDATURA\_id sarcina\_FK – atribut „id sarcina”
  - Referențiază o coloană din tabelul SARCINA (id sarcina)

❖ Constângeri de tip **NOT NULL**

- CANDIDATURA\_data\_inscriere\_NN – atribut „data\_inscriere”
  - Data la care își depune candidatura o echipă trebuie consemnată. (pe baza acestuia se vor calcula pozițiile candidaturii echipelor la o sarcină, ce va ajuta la stabilirea numărului de voturi)

#### 4.2.9. Tabelul SCRUTIN

❖ Constângeri de tip **PRIMARY KEY**

- SCRUTIN\_id\_scrutin\_PK – atribut „id\_scrutin”

❖ Constângeri de tip **FOREIGN KEY**

- SCRUTIN\_id\_sarcina\_FK – atribut „id\_sarcina”
  - Referențiază o coloană din tabelul SARCINA (id\_sarcina)

❖ Constângeri de tip **NOT NULL**

- SCRUTIN\_id\_sarcina\_NN - atribut „id\_sarcina”
  - Scrutinul trebuie să aibă o sarcină în mod obligatoriu
- SCRUTIN\_data\_inceput\_NN - atribut „data\_inceput\_inscriere”
  - Pentru un scrutin trebuie să existe o dată de începere a înscrierilor echipelor
- SCRUTIN\_data\_finalizare\_NN – atribut „data\_finalizare\_inscriere”
  - Pentru un scrutin trebuie să existe o dată de finalizare a înscrierilor echipelor
- SCRUTIN\_data\_scrutin\_NN - atribut „data\_scrutin”
  - Scrutinul trebuie să aibă o dată fixă în care se desfășoară
- SCRUTIN\_token\_scrutin\_NN - atribut „token\_scrutin”
  - Scrutinul trebuie să aibă o parolă pentru a permite alegătorilor să se autentifice în vederea exprimării votului

❖ Constângeri de tip **UNIQUE**

- SCRUTIN\_token\_scrutin\_UK - atribut „token\_scrutin”
  - Fiecare scrutin trebuie să aibă o parolă unică pentru a permite alegătorilor să se autentifice doar pentru un scrutin, în vederea exprimării votului pentru o anumită sarcină
- ❖ Constângeri de tip **CHECK**
  - SCRUTIN\_data\_finalizare\_CH - atribut „data\_finalizare\_inscriere”
    - Verific dacă data finalizării înscrierilor pentru un scrutin este strict mai mare decât data începerii înscrierilor
  - SCRUTIN\_data\_scrutin\_CH - atribut „data\_scrutin”
    - Verific dacă data desfășurării scrutinului este strict mai mare decât data finalizării înscrierilor

#### 4.2.10. Tabelul VOT (tabel asociativ STUDENT-SCRUTIN)

- ❖ Constângeri de tip **PRIMARY KEY**
  - VOT\_PK – attribute: „nr\_matricol” și „token\_scrutin” (cheie primară compusă)
- ❖ Constângeri de tip **FOREIGN KEY**
  - VOT\_nr\_matricol\_FK – atribut „nr\_matricol”
    - Referențiază o coloană din tabelul STUDENT (nr\_matricol)
  - VOT\_token\_scrutin\_FK – atribut „token\_scrutin”
    - Referențiază o coloană din tabelul SCRUTIN (token\_scrutin)
- ❖ Constângeri de tip **NOT NULL**
  - VOT\_moment\_vot\_NN – atribut „moment\_vot”
    - Este obligatoriu ca momentul votului să fie consemnat (pentru informații suplimentare despre votul care a fost înregistrat)
  - VOT\_optiune\_votata\_NN – atribut „optiune\_votata”
    - Opțiunea de vot a studentului (caracterizată de poziția unei echipe) este obligatorie.

#### 4.3. Schemele relaționale

- Tabel **FIRMA** (id\_firma#, nume\_firma, email, telefon)
- Tabel **PROIECT** (id\_proiect#, id\_firma (FK), nume\_proiect, buget)
- Tabel **SARCINA** (id\_sarcina#, id\_proiect (FK), nume\_sarcina)
- Tabel **DEPARTAMENT** (id\_departament#, nume\_departament)
- Tabel **ECHIPA** (id echipa#, nume echipa)
- Tabel **STUDENT** (nr\_matricol#, id\_departament (FK), nume, prenume)
- Tabel asociativ **STUDENT\_ECHIPA** (id echipa# (FK), id\_student# (FK))
- Tabel asociativ **CANDIDATURA** (id echipa# (FK), id\_sarcina# (FK), data\_inscriere)
- Tabel **SCRUTIN** (id\_scrutin#, id\_sarcina (FK), data\_inceput\_inscriere, data\_finalizare\_inscriere, data\_scrutin, token\_scrutin)
- Tabel asociativ **VOT** (nr\_matricol# (FK), token\_scrutin# (FK), moment\_vot, optiune\_votata)

#### 4.4. Descrierea constrângerilor ON DELETE CASCADE / SET NULL

Pentru toate constrângerile de tip FOREIGN KEY din baza de date, am adăugat constrângerea **ON DELETE CASCADE**.

- PROIECT\_id\_firma\_FK
  - Dacă o firmă este ștearsă din baza de date, atunci toate proiectele propuse de acea firmă vor fi șterse.
- SARCINA\_id\_proiect\_FK
  - Dacă un proiect este șters din baza de date, atunci toate sarcinile atribuite acelui proiect vor fi șterse.
- STUDENT\_id\_departament\_FK
  - Dacă un departament este șters din baza de date, atunci toți studenții care făceau parte din acel departament vor fi șterși.
- STUDENT\_ECHIPA\_id echipa\_FK
  - Atunci când o echipă este ștearsă din tabelul ECHIPA, ea va fi ștearsă și din lista de echipe și studenți în care apărea.
- STUDENT\_ECHIPA\_id\_student\_FK



- Atunci când un student este șters din tabelul STUDENT, el va fi șters și din lista de studenți și echipe în care apărea.
- CANDIDATURA\_id echipa\_FK
  - Atunci când o echipă este ștersă din tabelul ECHIPA, ea va fi ștersă și din lista de echipe și sarcini la care candida.
- CANDIDATURA\_id sarcina\_FK
  - Atunci când o sarcină este ștersă din tabelul SARCINA, ea va fi ștersă și din lista de sarcini și echipe pentru care se putea candida.
- SCRUTIN\_id sarcina\_FK
  - Dacă o sarcină este ștersă din baza de date, atunci toate scrutinele inițiate pentru acea sarcină vor fi șterse.
- VOT\_nr\_matricol\_FK
  - Atunci când un student este șters din tabelul STUDENT, toate înregistrările care conțin numărul matricol al studentului respectiv vor fi șterse din lista de studenți care au votat pentru un anumit scrutin.
- VOT\_token\_scrutin\_FK
  - Atunci când un scrutin este șters din tabelul SCRUTIN, el va fi șters și din lista de scrutine pentru care studenții au votat.

## 5. Script

### 5.1. Prima etapa: **DROP SEQUENCIES, ALTER TABLES** și **DROP TABLES**

**DROP SEQUENCE** FIRMA\_id\_firma\_SEQ;

**DROP SEQUENCE** PROIECT\_id\_proiect\_SEQ;

**DROP SEQUENCE** SACINA\_id\_sarcina\_SEQ;

**DROP SEQUENCE** DEPARTAMENT\_id\_departament\_SEQ;

**DROP SEQUENCE** ECHIPA\_id echipa\_SEQ;

**DROP SEQUENCE** STUDENT\_nr\_matricol\_SEQ;

**DROP SEQUENCE** SCRUTIN\_id\_token\_SEQ;

**DROP SEQUENCE** SCRUTIN\_id\_scrutin\_SEQ;

**ALTER TABLE** VOT

**DROP CONSTRAINT** VOT\_nr\_matricol\_FK

**DROP CONSTRAINT** VOT\_token\_scrutin\_FK;

**DROP TABLE** VOT;

**ALTER TABLE** SCRUTIN

**DROP CONSTRAINT** SCRUTIN\_id\_sarcina\_FK;

**DROP TABLE** SCRUTIN;

**ALTER TABLE** CANDIDATURA

**DROP CONSTRAINT** CANDIDATURA\_id echipa\_FK

**DROP CONSTRAINT** CANDIDATURA\_id\_sarcina\_FK;

**DROP TABLE** CANDIDATURA;

**ALTER TABLE** STUDENT\_ECHIPA

```

DROP CONSTRAINT STUDENT_ECHIPA_id_echipa_FK
DROP CONSTRAINT STUDENT_ECHIPA_id_student_FK;
DROP TABLE STUDENT_ECHIPA;
ALTER TABLE STUDENT
    DROP CONSTRAINT STUDENT_id_departament_FK;
DROP TABLE STUDENT;
DROP TABLE ECHIPA;
DROP TABLE DEPARTAMENT;
ALTER TABLE SARCINA
    DROP CONSTRAINT SARCINA_id_proiect_FK;
DROP TABLE SARCINA;
ALTER TABLE PROIECT
    DROP CONSTRAINT PROIECT_id_firma_FK;
DROP TABLE PROIECT;
DROP TABLE FIRMA;

```

## 5.2. Etapa a doua: Crearea secvențelor, a tabelelor și a constrângerilor

### 5.2.1. Tabelul **FIRMA**

```

CREATE SEQUENCE FIRMA_id_firma_SEQ
START WITH 1
MAXVALUE 999
NOCYCLE NOCACHE;

CREATE TABLE FIRMA(

```

```

id_firma NUMBER(3,0)

CONSTRAINT FIRMA_id_firma_PK PRIMARY KEY,

nume_firma VARCHAR2(100 CHAR)

CONSTRAINT FIRMA_nume_firma_NN NOT NULL

CONSTRAINT FIRMA_nume_firma_UK UNIQUE,

email VARCHAR2(40 CHAR)

CONSTRAINT FIRMA_email_NN NOT NULL

CONSTRAINT FIRMA_email_UK UNIQUE

CONSTRAINT FIRMA_email_CH CHECK(REGEXP_LIKE(email,
'[:alnum:]]+@[[:alnum:]]+\.[[:alnum:]]')),

telefon VARCHAR2(10 CHAR)

CONSTRAINT FIRMA_telefon_NN NOT NULL

CONSTRAINT FIRMA_telefon_UK UNIQUE

CONSTRAINT FIRMA_telefon_CH CHECK(LENGTH(telefon) = 10)

);

```

### 5.2.2. Tabelul **PROIECT**

```

CREATE SEQUENCE PROIECT_id_proiect_SEQ

START WITH 1

MAXVALUE 9999

NOCYCLE NOCACHE;

CREATE TABLE PROIECT(

id_proiect NUMBER(4,0)

```

```

        CONSTRAINT PROIECT_id_proiect_PK PRIMARY KEY,

id_firma NUMBER(3,0)

        CONSTRAINT PROIECT_id_firma_FK REFERENCES FIRMA(id_firma) ON
DELETE CASCADE

        CONSTRAINT PROIECT_id_firma_NN NOT NULL,

nume_proiect VARCHAR2(80 CHAR)

        CONSTRAINT PROIECT_nume_proiect_NN NOT NULL,

buget NUMBER(4,0)

        CONSTRAINT PROIECT_buget_NN NOT NULL

);

```

### 5.2.3. Tabelul **SARCINA**

```

CREATE SEQUENCE SACINA_id_sarcina_SEQ

START WITH 1

MAXVALUE 99999

NOCYCLE NOCACHE;

```

```

CREATE TABLE SARCINA(

    id_sarcina NUMBER(5,0)

        CONSTRAINT SARCINA_id_sarcina_PK PRIMARY KEY,

    id_proiect NUMBER(4,0)

        CONSTRAINT SARCINA_id_proiect_FK REFERENCES PROIECT(id_proiect) ON
DELETE CASCADE

        CONSTRAINT SARCINA_id_proiect_NN NOT NULL,

    nume_sarcina VARCHAR2(100 CHAR)

```

```
CONSTRAINT SARCINA_num_sarcina_NN NOT NULL  
);
```

#### 5.2.4. Tabelul **DEPARTAMENT**

```
CREATE SEQUENCE DEPARTAMENT_id_department_SEQ  
INCREMENT BY 10  
START WITH 10  
MAXVALUE 990  
NOCYCLE NOCACHE;
```

```
CREATE TABLE DEPARTAMENT(  
    id_department NUMBER(3,0)  
        CONSTRAINT DEPARTAMENT_id_dep_PK PRIMARY KEY,  
    nume_department VARCHAR2(30 CHAR)  
        CONSTRAINT DEPARTAMENT_nume_dep_NN NOT NULL  
        CONSTRAINT DEPARTAMENT_nume_dep_UK UNIQUE  
);
```

#### 5.2.5. Tabelul **ECHIPA**

```
CREATE SEQUENCE ECHIPA_id_echipa_SEQ  
START WITH 1  
MAXVALUE 9999  
NOCYCLE NOCACHE;
```

```

CREATE TABLE ECHIPA(
    id_echipa NUMBER(4,0)
        CONSTRAINT ECHIPA_id_echipa_PK PRIMARY KEY,
    nume_echipa VARCHAR2(60 CHAR)
        CONSTRAINT ECHIPA_nume_echipa_NN NOT NULL
        CONSTRAINT ECHIPA_nume_echipa_UK UNIQUE
);

```

#### 5.2.6. Tabelul **STUDENT**

```

CREATE SEQUENCE STUDENT_nr_matricol_SEQ
INCREMENT BY 1
START WITH 10000
MAXVALUE 99999
NOCYCLE NOCACHE;

```

```

CREATE TABLE STUDENT(
    nr_matricol NUMBER(5,0)
        CONSTRAINT STUDENT_nr_matricol_PK PRIMARY KEY,
    id_departament NUMBER(3,0)
        CONSTRAINT STUDENT_id_departament_FK REFERENCES
DEPARTAMENT(id_departament) ON DELETE CASCADE
        CONSTRAINT STUDENT_id_departament_NN NOT NULL,
    nume VARCHAR2(40 CHAR)
        CONSTRAINT STUDENT_nume_student_NN NOT NULL,

```

prenume VARCHAR2(50 CHAR)

**CONSTRAINT** STUDENT\_prenume\_student\_NN **NOT NULL**

);

#### 5.2.7. Tabelul asociativ **STUDENT\_ECHIPA**

**CREATE TABLE** STUDENT\_ECHIPA(

id\_echipa NUMBER(4,0)

**CONSTRAINT** STUDENT\_ECHIPA\_id\_echipa\_FK REFERENCES  
ECHIPA(id\_echipa) **ON DELETE CASCADE**,

id\_student NUMBER(5,0)

**CONSTRAINT** STUDENT\_ECHIPA\_id\_student\_FK REFERENCES  
STUDENT(nr\_matricol) **ON DELETE CASCADE**,

**CONSTRAINT** STUDENT\_ECHIPA\_PK **PRIMARY KEY**(id\_echipa, id\_student)

);

#### 5.2.8. Tabelul asociativ **CANDIDATURA**

**CREATE TABLE** CANDIDATURA(

id\_echipa NUMBER(4,0)

**CONSTRAINT** CANDIDATURA\_id\_echipa\_FK REFERENCES ECHIPA(id\_echipa)  
**ON DELETE CASCADE**,

id\_sarcina NUMBER(5,0)

**CONSTRAINT** CANDIDATURA\_id\_sarcina\_FK REFERENCES  
SARCINA(id\_sarcina) **ON DELETE CASCADE**,

**CONSTRAINT** CANDIDATURA\_PK **PRIMARY KEY**(id\_echipa, id\_sarcina),

data\_inscriere DATE



**CONSTRAINT** CANDIDATURA\_data\_inscriere\_NN **NOT NULL**

);

#### 5.2.9. Tabelul **SCRUTIN**

**CREATE SEQUENCE** SCRUTIN\_id\_scrutin\_SEQ

**START WITH** 1

**MAXVALUE** 999999

**NOCYCLE NOCACHE;**

**CREATE SEQUENCE** SCRUTIN\_id\_token\_SEQ

**INCREMENT BY** 1

**START WITH** 1000000

**MAXVALUE** 9999999

**NOCYCLE NOCACHE;**

**CREATE TABLE** SCRUTIN(  
id\_scrutin **NUMBER**(6,0)

**CONSTRAINT** SCRUTIN\_id\_scrutin\_PK **PRIMARY KEY**,

id\_sarcina **NUMBER**(5,0)

**CONSTRAINT** SCRUTIN\_id\_sarcina\_FK **REFERENCES** SARCINA(id\_sarcina) **ON DELETE CASCADE**

**CONSTRAINT** SCRUTIN\_id\_sarcina\_NN **NOT NULL**,

data\_inceput\_inscriere **DATE**

**CONSTRAINT** SCRUTIN\_data\_inceput\_NN **NOT NULL**,

```

data_finalizare_inscriere DATE

    CONSTRAINT SCRUTIN_data_finalizare_NN NOT NULL,

data_scrutin DATE

    CONSTRAINT SCRUTIN_data_scrutin_NN NOT NULL,

token_scrutin NUMBER(7,0)

    CONSTRAINT SCRUTIN_token_scrutin_NN NOT NULL

    CONSTRAINT SCRUTIN_token_scrutin_UK UNIQUE,

    CONSTRAINT SCRUTIN_data_finalizare_CH CHECK (data_finalizare_inscriere >
data_inceput_inscriere),

    CONSTRAINT SCRUTIN_data_scrutin_CH CHECK (data_scrutin >
data_finalizare_inscriere)

);

```

#### 5.2.10. Tabelul asociativ **VOT**

```

CREATE TABLE VOT(

    nr_matricol NUMBER(5,0)

        CONSTRAINT VOT_nr_matricol_FK REFERENCES STUDENT(nr_matricol) ON
DELETE CASCADE,

    token_scrutin NUMBER(7,0)

        CONSTRAINT VOT_token_scrutin_FK REFERENCES SCRUTIN(token_scrutin) ON
DELETE CASCADE,

    CONSTRAINT VOT_PK PRIMARY KEY(nr_matricol, token_scrutin),

    moment_vot DATE

        CONSTRAINT VOT_moment_vot_NN NOT NULL,

    optiune_votata NUMBER(4,0)

```

**CONSTRAINT VOT\_optiune\_votata\_NN NOT NULL**

);

### 5.3. Ultima etapa: Introducerea datelor

- Am adăugat câteva exemple de populări pentru fiecare tabel din baza de date.

#### 5.3.1. Tabelul **FIRMA**

**INSERT INTO** FIRMA(id\_firma, nume\_firma, email, telefon)

**VALUES** (FIRMA\_id\_firma\_SEQ.NEXTVAL, 'ENDAVA ROMANIA SRL',  
'informatii@endava.ro', '0726579800');

**INSERT INTO** FIRMA(id\_firma, nume\_firma, email, telefon)

**VALUES** (FIRMA\_id\_firma\_SEQ.NEXTVAL, 'ORACLE ROMANIA SRL',  
'informatii@oracle.ro', '0726456789');

**INSERT INTO** FIRMA(id\_firma, nume\_firma, email, telefon)

**VALUES** (FIRMA\_id\_firma\_SEQ.NEXTVAL, 'METRO SYSTEMS ROMANIA SRL',  
'informatii@metrosystems.ro', '0783559192');

**INSERT INTO** FIRMA(id\_firma, nume\_firma, email, telefon)

**VALUES** (FIRMA\_id\_firma\_SEQ.NEXTVAL, 'AROBS TRANSILVANIA SOFTWARE  
SRL', 'informatii@arobs.ro', '0786862006');

**INSERT INTO** FIRMA(id\_firma, nume\_firma, email, telefon)

**VALUES** (FIRMA\_id\_firma\_SEQ.NEXTVAL, 'QUALITEST DC RO SRL',  
'informatii@qualitest.ro', '0702867013');

### 5.3.2. Tabelul **PROIECTE**

```
INSERT INTO PROIECT(id_proiect, id_firma, nume_proiect, buget)
```

```
VALUES (PROIECT_id_proiect_SEQ.NEXTVAL, 1, 'Brat robotic', 800);
```

```
INSERT INTO PROIECT(id_proiect, id_firma, nume_proiect, buget)
```

```
VALUES (PROIECT_id_proiect_SEQ.NEXTVAL, 2, 'Sistem de vot electronic', 1000);
```

```
INSERT INTO PROIECT(id_proiect, id_firma, nume_proiect, buget)
```

```
VALUES (PROIECT_id_proiect_SEQ.NEXTVAL, 3, 'Joc video cu tematica militara', 1500);
```

```
INSERT INTO PROIECT(id_proiect, id_firma, nume_proiect, buget)
```

```
VALUES (PROIECT_id_proiect_SEQ.NEXTVAL, 4, 'Sistem directie de zbor avioane',  
900);
```

```
INSERT INTO PROIECT(id_proiect, id_firma, nume_proiect, buget)
```

```
VALUES (PROIECT_id_proiect_SEQ.NEXTVAL, 5, 'Aplicatie pentru comertul local', 400);
```

### 5.3.3. Tabelul **SARCINA**

```
INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES  
(SARCINA_id_sarcina_SEQ.NEXTVAL, 1, 'DESIGN');
```

```
INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES  
(SARCINA_id_sarcina_SEQ.NEXTVAL, 1, 'CALCULE MATEMATICE');
```

```
INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES  
(SARCINA_id_sarcina_SEQ.NEXTVAL, 1, 'PROGRAMARE IN JAVA');
```

```

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 1, 'ACHIZITIE COMPONENTE');

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 1, 'ASAMBLARE COMPONENTE');

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 2, 'PROIECTARE BAZA DE DATE');

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 2, 'PROGRAMARE IN PHP');

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 3, 'PROGRAMARE IN PYTHON');

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 3, 'DESIGN');

INSERT INTO SARCINA(id_sarcina, id_proiect, nume_sarcina) VALUES
(SACINA_id_sarcina_SEQ.NEXTVAL, 4, 'CALCULE MATEMATICE');

```

#### 5.3.4. Tabelul **DEPARTAMENT**

```

INSERT INTO DEPARTAMENT(id_departament, nume_departament) VALUES
(DEPARTAMENT_id_departament_SEQ.NEXTVAL, 'HUMAN RESOURCES');

INSERT INTO DEPARTAMENT(id_departament, nume_departament) VALUES
(DEPARTAMENT_id_departament_SEQ.NEXTVAL, 'EDUCATIONAL');

INSERT INTO DEPARTAMENT(id_departament, nume_departament) VALUES
(DEPARTAMENT_id_departament_SEQ.NEXTVAL, 'DESIGN');

INSERT INTO DEPARTAMENT(id_departament, nume_departament) VALUES
(DEPARTAMENT_id_departament_SEQ.NEXTVAL, 'PUBLIC RELATIONSHIPS');

INSERT INTO DEPARTAMENT(id_departament, nume_departament) VALUES
(DEPARTAMENT_id_departament_SEQ.NEXTVAL, 'MANAGEMENT AND
FUNDRAISING');

```

#### 5.3.5. Tabelul **STUDENT**

```
INSERT INTO STUDENT(nr_matricol, id_departament, nume, prenume) VALUES  
(STUDENT_nr_matricol_SEQ.NEXTVAL, 10, 'Popescu', 'Mihai');
```

```
INSERT INTO STUDENT(nr_matricol, id_departament, nume, prenume) VALUES  
(STUDENT_nr_matricol_SEQ.NEXTVAL, 10, 'Popescu', 'Mihail');
```

```
INSERT INTO STUDENT(nr_matricol, id_departament, nume, prenume) VALUES  
(STUDENT_nr_matricol_SEQ.NEXTVAL, 30, 'Georgescu', 'Catalin');
```

```
INSERT INTO STUDENT(nr_matricol, id_departament, nume, prenume) VALUES  
(STUDENT_nr_matricol_SEQ.NEXTVAL, 40, 'Alexandru', 'Ana');
```

```
INSERT INTO STUDENT(nr_matricol, id_departament, nume, prenume) VALUES  
(STUDENT_nr_matricol_SEQ.NEXTVAL, 40, 'Decu', 'Madalina');
```

#### 5.3.6. Tabelul **ECHIPA**

```
INSERT INTO ECHIPA(id echipa, nume echipa) VALUES  
(ECHIPA_id echipa_SEQ.NEXTVAL, 'Vulturii Albastri');
```

```
INSERT INTO ECHIPA(id echipa, nume echipa) VALUES  
(ECHIPA_id echipa_SEQ.NEXTVAL, 'Testoasele Ninja');
```

```
INSERT INTO ECHIPA(id echipa, nume echipa) VALUES  
(ECHIPA_id echipa_SEQ.NEXTVAL, 'Cangurii saltareti');
```

```
INSERT INTO ECHIPA(id echipa, nume echipa) VALUES  
(ECHIPA_id echipa_SEQ.NEXTVAL, 'Pantere pe plaja');
```

```
INSERT INTO ECHIPA(id echipa, nume echipa) VALUES  
(ECHIPA_id echipa_SEQ.NEXTVAL, 'Girafele Dure');
```

#### 5.3.7. Tabelul asociativ **STUDENT\_ECHIPA**

```
INSERT INTO STUDENT_ECHIPA(id echipa, id student) VALUES (2, 10001);
```

INSERT INTO STUDENT\_ECHIPA(id echipa, id\_student) VALUES (38, 10002);

INSERT INTO STUDENT\_ECHIPA(id echipa, id\_student) VALUES (1, 10003);

INSERT INTO STUDENT\_ECHIPA(id echipa, id\_student) VALUES (1, 10004);

INSERT INTO STUDENT\_ECHIPA(id echipa, id\_student) VALUES (2, 10005);

#### 5.3.8. Tabelul asociativ **CANDIDATURA**

INSERT INTO CANDIDATURA(id echipa, id\_sarcina, data\_inscriere) VALUES (1, 1, TO\_DATE('31-03-2021 12:23:35', 'DD-MM-YYYY HH24:MI:SS'));

INSERT INTO CANDIDATURA(id echipa, id\_sarcina, data\_inscriere) VALUES (2, 1, TO\_DATE('31-03-2021 14:25:54', 'DD-MM-YYYY HH24:MI:SS'));

INSERT INTO CANDIDATURA(id echipa, id\_sarcina, data\_inscriere) VALUES (3, 1, TO\_DATE('01-03-2021 21:34:35', 'DD-MM-YYYY HH24:MI:SS'));

INSERT INTO CANDIDATURA(id echipa, id\_sarcina, data\_inscriere) VALUES (18, 1, TO\_DATE('01-03-2021 23:24:54', 'DD-MM-YYYY HH24:MI:SS'));

INSERT INTO CANDIDATURA(id echipa, id\_sarcina, data\_inscriere) VALUES (21, 1, TO\_DATE('02-03-2021 02:01:23', 'DD-MM-YYYY HH24:MI:SS'));

#### 5.3.9. Tabelul **SCRUTIN**

INSERT INTO SCRUTIN(id\_scrutin, id\_sarcina, data\_inceput\_inscriere, data\_finalizare\_inscriere, data\_scrutin, token\_scrutin)

VALUES (SCRUTIN\_id\_scrutin\_SEQ.NEXTVAL, 1, TO\_DATE('31-03-2021', 'DD-MM-YYYY'), TO\_DATE('02-04-2021', 'DD-MM-YYYY'), TO\_DATE('03-04-2021', 'DD-MM-YYYY'), SCRUTIN\_id\_token\_SEQ.NEXTVAL);

INSERT INTO SCRUTIN(id\_scrutin, id\_sarcina, data\_inceput\_inscriere, data\_finalizare\_inscriere, data\_scrutin, token\_scrutin)

```
VALUES (SCRUTIN_id_scrutin_SEQ.NEXTVAL, 2, TO_DATE('31-03-2021', 'DD-MM-YYYY'), TO_DATE('02-04-2021', 'DD-MM-YYYY'), TO_DATE('03-04-2021', 'DD/MM/YYYY'), SCRUTIN_id_token_SEQ.NEXTVAL);
```

```
INSERT INTO SCRUTIN(id_scrutin, id_sarcina, data_inceput_inscriere, data_finalizare_inscriere, data_scrutin, token_scrutin)
```

```
VALUES (SCRUTIN_id_scrutin_SEQ.NEXTVAL, 3, TO_DATE('31-03-2021', 'DD-MM-YYYY'), TO_DATE('02-04-2021', 'DD-MM-YYYY'), TO_DATE('03-04-2021', 'DD-MM-YYYY'), SCRUTIN_id_token_SEQ.NEXTVAL);
```

```
INSERT INTO SCRUTIN(id_scrutin, id_sarcina, data_inceput_inscriere, data_finalizare_inscriere, data_scrutin, token_scrutin)
```

```
VALUES (SCRUTIN_id_scrutin_SEQ.NEXTVAL, 4, TO_DATE('31-03-2021', 'DD-MM-YYYY'), TO_DATE('02-04-2021', 'DD-MM-YYYY'), TO_DATE('03-04-2021', 'DD-MM-YYYY'), SCRUTIN_id_token_SEQ.NEXTVAL);
```

```
INSERT INTO SCRUTIN(id_scrutin, id_sarcina, data_inceput_inscriere, data_finalizare_inscriere, data_scrutin, token_scrutin)
```

```
VALUES (SCRUTIN_id_scrutin_SEQ.NEXTVAL, 5, TO_DATE('31-03-2021', 'DD-MM-YYYY'), TO_DATE('02-04-2021', 'DD-MM-YYYY'), TO_DATE('03-04-2021', 'DD-MM-YYYY'), SCRUTIN_id_token_SEQ.NEXTVAL);
```

#### 5.3.10. Tabelul asociativ **VOT**

```
INSERT INTO VOT(nr_matricol, token_scrutin, moment_vot, optiune_votata) VALUES (10000, '1000000', TO_DATE('03-04-2021 12:23:32', 'DD-MM-YYYY HH24:MI:SS'), 2);
```

```
INSERT INTO VOT(nr_matricol, token_scrutin, moment_vot, optiune_votata) VALUES (10000, '1000002', TO_DATE('03-04-2021 11:13:22', 'DD-MM-YYYY HH24:MI:SS'), 2);
```



```
INSERT INTO VOT(nr_matricol, token_scrutin, moment_vot, optiune_votata) VALUES  
(10000, '1000003', TO_DATE('03-04-2021 04:15:21', 'DD-MM-YYYY HH24:MI:SS'), 2);
```

```
INSERT INTO VOT(nr_matricol, token_scrutin, moment_vot, optiune_votata) VALUES  
(10000, '1000004', TO_DATE('03-04-2021 06:18:44', 'DD-MM-YYYY HH24:MI:SS'), 1);
```

```
INSERT INTO VOT(nr_matricol, token_scrutin, moment_vot, optiune_votata) VALUES  
(10000, '1000005', TO_DATE('21-05-2021 09:55:21', 'DD-MM-YYYY HH24:MI:SS'), 1);
```