

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

**PROIECT**  
**BAZE DE DATE**

PROFESOR COORDONATOR:

VASILE SILVIU-LAURENȚIU

STUDENT:

DINOIU NICOLETA-ANASTASIA

BUCUREȘTI

2022

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

**GESTIONAREA BAZEI DE DATE A UNEI FARMACII**

PROFESOR COORDONATOR:

VASILE SILVIU-LAURENȚIU

STUDENT:

DINOIU NICOLETA-ANASTASIA

BUCUREȘTI

2022

## CUPRINS

1. Prezentarea modelului.....	5
2. Regulile modelului.....	6
3. Diagrama Entitate-Relație.....	7
3.1. Reprezentarea diagramei.....	7
3.2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților.....	8
3.2.1. Descrierea entităților, atributelor și a cheilor.....	8
3.2.1.1. Tabelul POSTURI .....	8
3.2.1.2. Tabelul ANGAJATI .....	8
3.2.1.3. Tabelul PACIENTI.....	9
3.2.1.4. Tabelul MEDICI_PRESCRIPTORI.....	9
3.2.1.5. Tabelul DIAGNOSTICE .....	10
3.2.1.6. Tabelul RETETE .....	10
3.2.1.7. Tabelul MEDICAMENTE .....	11
3.2.1.8. Tabelul PRODUCATORI.....	11
3.2.2. Descrierea relațiilor și a cardinalităților.....	12
3.2.2.1. POSTURI – ANGAJATI.....	12
3.2.2.2. ANGAJATI – RETETE.....	12
3.2.2.3. PACIENTI – RETETE .....	13
3.2.2.4. MEDICI_PRESCRIPTORI – RETETE .....	13
3.2.2.5. DIAGNOSTICE – RETETE .....	14
3.2.2.6. RETETE – MEDICAMENTE.....	14
3.2.2.7. MEDICAMENTE – PRODUCATORI .....	15
4. Diagrama conceptuală.....	16
4.1. Reprezentarea diagramei .....	16

4.2.	Descrierea constrângerilor de integritate.....	17
4.2.1.	Tabelul POSTURI.....	17
4.2.2.	Tabelul ANGAJATI.....	17
4.2.3.	Tabelul PACIENTI .....	18
4.2.4.	Tabelul MEDICI_PRESCRIPTORI .....	19
4.2.5.	Tabelul DIAGNOSTICE .....	20
4.2.6.	Tabelul RETETE .....	20
4.2.7.	Tabelul RETETE_MEDICAMENTE.....	21
4.2.8.	Tabelul MEDICAMENTE.....	21
4.2.9.	Tabelul PRODUCATORI.....	22
4.3.	Schemele relaționale .....	23
4.3.1.	Schemele relaționale .....	23
4.3.2.	Descrierea constrângerilor ON DELETE CASCADE/SET NULL.....	24
5.	Implementarea într-un SGBD.....	25
5.1.	Introducere .....	25
5.2.	Crearea tabelelor și a constrângerilor.....	25
5.2.1.	Crearea tabelului POSTURI .....	25
5.2.2.	Crearea tabelului ANGAJATI .....	25
5.2.3.	Crearea tabelului PACIENTI.....	26
5.2.4.	Crearea tabelului MEDICI_PRESCRIPTORI .....	27
5.2.5.	Crearea tabelului DIAGNOSTICE .....	27
5.2.6.	Crearea tabelului RETETE .....	28
5.2.7.	Crearea tabelului PRODUCATORI.....	29
5.2.8.	Crearea tabelului MEDICAMENTE.....	29
5.2.9.	Crearea tabelului RETETE_MEDICAMENTE.....	30

5.3.	Introducerea datelor.....	30
5.3.1.	Introducerea datelor în tabelul POSTURI.....	30
5.3.2.	Introducerea datelor în tabelul ANGAJATI .....	31
5.3.3.	Introducerea datelor în tabelul PACIENTI.....	32
5.3.4.	Introducerea datelor în tabelul MEDICI_PRESCRIPTORI .....	32
5.3.5.	Introducerea datelor în tabelul DIAGNOSTICE .....	33
5.3.6.	Introducerea datelor în tabelul RETETE .....	33
5.3.7.	Introducerea datelor în tabelul PRODUCATORI.....	34
5.3.8.	Introducerea datelor în tabelul MEDICAMENTE.....	34
5.3.9.	Introducerea datelor în tabelul RETETE_MEDICAMENTE .....	35

## **1. Prezentarea modelului**

Tema aleasă pentru proiectul la disciplina Baze de date este „Gestionarea bazei de date a unei farmacii”. Farmacia, la modul general, face referire la locul unde se eliberează medicamentele conform rețetelor medicale și normelor din farmacologie.

Proiectul are în vedere managementul unei singure farmacii dedicate exclusiv pacienților care aduc rețete eliberate de diferiți medici prescriptori care, pe baza unor diagnostice, recomandă medicamente adecvate fiecărui pacient, pe fiecare rețetă fiind menționat un singur diagnostic.

Angajații ocupă diverse posturi și asigură buna funcționare a farmaciei. Dintre aceștia, asistenții, farmaciștii și farmacistul șef al farmaciei se ocupă de gestionarea rețetelor.

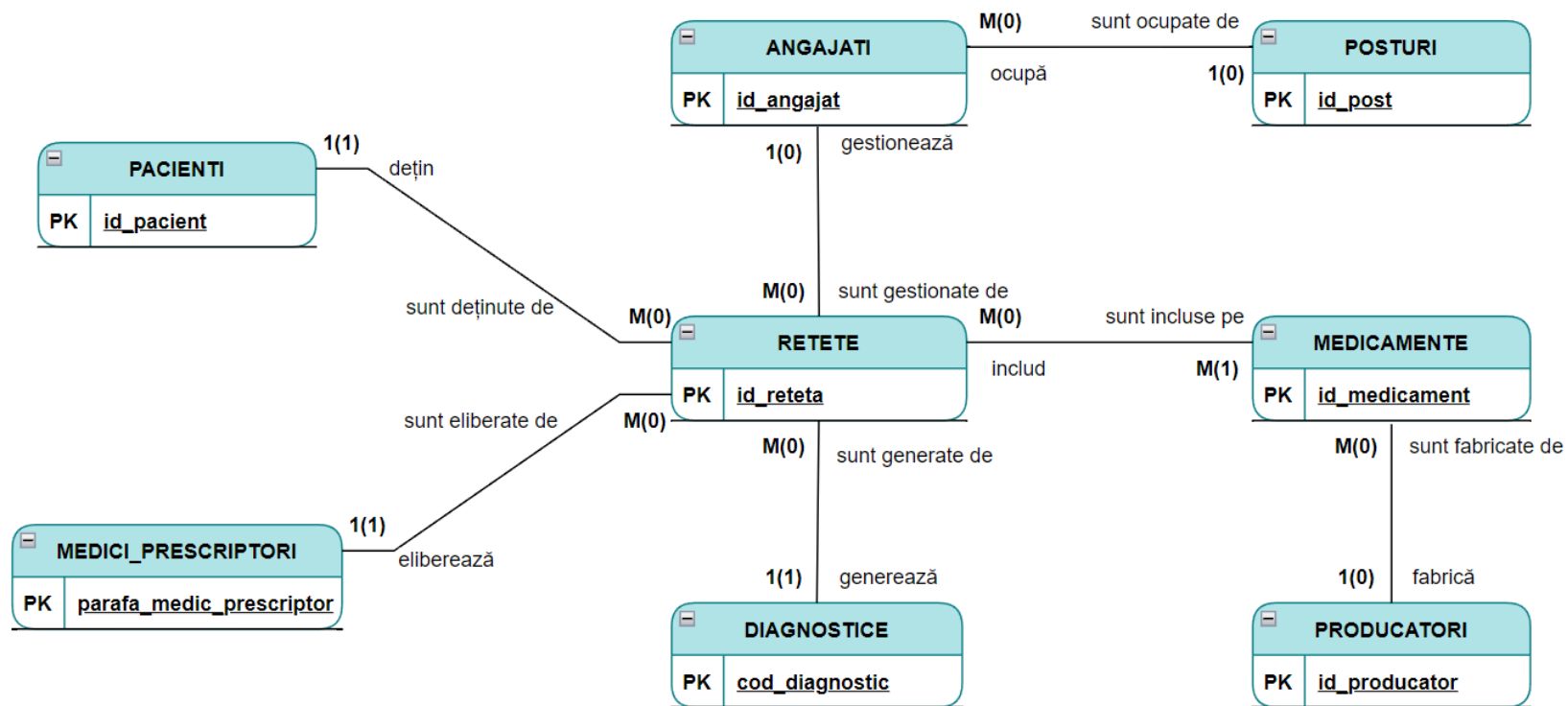
Toate medicamentele prescrise pe o rețetă vor fi achiziționate integral, o singură dată, iar fiecare medicament existent în farmacia descrisă va avea câte un producător.

## 2. Regulele modelului

- Pot exista angajați pentru care nu este alocat niciun post la un anumit moment dat, dar și angajați care ocupă un singur post.  $1(0)$
- Există posturi fără angajați, dar și posturi care au mai mulți angajați.  $M(0)$
- Pot exista angajați care nu gestionează rețete, precum un îngrijitor sau un agent de pază, dar și angajați care gestionează mai multe rețete.  $M(0)$
- Pot exista rețete care să nu fie gestionate de niciun angajat la un anumit moment dat, dar și rețete care să fie gestionate de un singur angajat.  $1(0)$
- Pe o rețetă este trecut un singur pacient.  $1(1)$
- Pot exista pacienți pentru care nu este înregistrată nicio rețetă, dar și pacienți care au mai multe rețete.  $M(0)$
- O rețetă este generată de un singur diagnostic.  $1(1)$
- Pot exista diagnostice pentru care nu este înregistrată nicio rețetă, dar și diagnostice care au mai multe rețete.  $M(0)$
- Pe o rețetă este trecut un singur medic prescriptor.  $1(1)$
- Pot exista medici prescriptori care nu au eliberat nicio rețetă, dar și medici prescriptori care eliberează mai multe rețete.  $M(0)$
- O rețetă trebuie să includă cel puțin un medicament.  $M(1)$
- Pot exista medicamente care să nu fie incluse pe nicio rețetă, iar un medicament poate apărea pe mai multe rețete.  $M(0)$
- Pot exista medicamente care să nu aibă niciun producător la un anumit moment dat, dar și medicamente care au un singur producător.  $1(0)$
- Pot exista producători care nu au fabricat momentan niciun medicament, dar și producători care fabrică mai multe medicamente.  $M(0)$

### 3. Diagrama Entitate-Relație

#### 3.1. Reprezentarea diagramei





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

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

##### 3.2.1.1. Tabelul *POSTURI*

Tabelul POSTURI stochează informații despre posturile pe care le pot ocupa angajații farmaciei. Structura tabelului POSTURI este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	id_post	identificatorul unic al postului	DECIMAL(5, 0)
	denumire_post	denumirea postului (îngrijitor, farmacist etc.)	VARCHAR(20)
	salariu	salariul net al postului	DECIMAL(4, 0)

##### 3.2.1.2. Tabelul *ANGAJATI*

Tabelul ANGAJATI stochează date despre angajații care lucrează în farmacie. Structura tabelului ANGAJATI este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	id_angajat	identificatorul unic al angajatului	DECIMAL(5, 0)
	nume	date personale	VARCHAR(20)
	prenume	date personale	VARCHAR(30)
	cnp	date personale	DECIMAL(13, 0)
FK	id_post	postul ocupat de către angajat	DECIMAL(5, 0)
	email	date de contact	VARCHAR(30)
	telefon	date de contact	VARCHAR(10)

### 3.2.1.3. Tabelul *PACIENTI*

Tabelul PACIENTI stochează date despre pacienții farmaciei. Structura tabelului PACIENTI este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	id_pacient	identificatorul unic al pacientului	DECIMAL(5, 0)
	nume	date personale	VARCHAR(20)
	prenume	date personale	VARCHAR(30)
	cnp	date personale	DECIMAL(13, 0)

### 3.2.1.4. Tabelul *MEDICI\_PRESCRIPTORI*

Tabelul MEDICI\_PRESCRIPTORI stochează date despre medicii care prescriu rețetele pacienților farmaciei. Structura tabelului MEDICI\_PRESCRIPTORI este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	parafa_medic_prescriptor	identificatorul unic al medicului	DECIMAL(5, 0)
	unitate_medicala	unitatea medicală din care face parte medicul	VARCHAR(50)
	nume	date personale	VARCHAR(20)
	prenume	date personale	VARCHAR(30)
	email	date de contact	VARCHAR(30)
	telefon	date de contact	VARCHAR(10)

### 3.2.1.5. Tabelul *DIAGNOSTICE*

Tabelul DIAGNOSTICE stochează date despre diferite diagnostice. Structura tabelului DIAGNOSTICE este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	cod_diagnostic	identificatorul unic al diagnosticului	DECIMAL(5, 0)
	denumire_diagnostic	denumirea diagnosticului	VARCHAR(100)

### 3.2.1.6. Tabelul *RETETE*

Tabelul RETETE stochează date despre prescripțiile medicale ale pacienților. Structura tabelului RETETE este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	id_reteta	identificatorul unic al rețetei	DECIMAL(5, 0)
FK	id_pacient	pacientul care deține rețeta	DECIMAL(5, 0)
FK	cod_diagnostic	diagnosticul pacientului	DECIMAL(5, 0)
	tip_tratament	tipul tratamentului (acut, subacut sau cronic)	VARCHAR(7)
FK	parafa_medic_prescriptor	medicul care a prescris rețeta	DECIMAL(5, 0)
	data_prescriere	data prescrierii rețetei	DATE
FK	id_angajat	angajatul care se ocupă de rețetă	DECIMAL(5, 0)

### 3.2.1.7. Tabelul *MEDICAMENTE*

Tabelul MEDICAMENTE stochează date despre medicamentele prezente în farmacie, care pot apărea pe rețetele pacienților. Structura tabelului MEDICAMENTE este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	id_medicament	identificatorul unic al medicamentului	DECIMAL(5, 0)
	dci	denumirea comună internațională	VARCHAR(50)
	denumire_comerciala	denumirea comercială a medicamentului	VARCHAR(30)
	forma_farmaceutica	forma farmaceutică a medicamentului (capsule, comprimate etc.)	VARCHAR(50)
	concentratie	cantitatea de substanță activă din compoziția medicamentului	VARCHAR(30)
FK	id_producator	producătorul medicamentului	DECIMAL(5, 0)

### 3.2.1.8. Tabelul *PRODUCATORI*

Tabelul PRODUCATORI stochează date despre producătorii de medicamente. Structura tabelului PRODUCATORI este următoarea:

Cheie	Denumire atribut	Descriere	Tipul de dată
PK	id_producator	identificatorul unic al diagnosticului	DECIMAL(5, 0)
	producator	numele producatorului de medicamente	VARCHAR(50)
	tara	țara în care se află producătorul	VARCHAR(20)

### 3.2.2. Descrierea relațiilor și a cardinalităților

#### 3.2.2.1. *POSTURI – ANGAJATI*

**Relația:** *POSTURI* sunt ocupate de *ANGAJATI*

**Cardinalități:**

- Cardinalitatea minimă:
  - Câți angajați trebuie să ocupe același post? => 0
    - pot exista posturi fără angajați
  - Câte posturi trebuie să fie ocupate de un angajat? => 0
    - în cazul în care un post este șters, angajatul rămâne în baza de date, dar cu atributul *id\_post* setat NULL
- Cardinalitatea maximă:
  - Câți angajați pot ocupa același post? => Mulți
  - Câte posturi pot fi ocupate de un angajat? => 1

#### 3.2.2.2. *ANGAJATI – RETETE*

**Relația:** *ANGAJATI* gestionează *RETETE*

**Cardinalități:**

- Cardinalitatea minimă:
  - Câte rețete trebuie să fie gestionate de un angajat? => 0
    - există și angajati care nu se ocupă de gestionarea rețetelor (un îngrijitor sau un agent de pază)
  - Câți angajați trebuie să gestioneze aceeași rețetă? => 0
    - în cazul în care un angajat este șters, rețeta rămâne în baza de date, dar cu atributul *id\_angajat* setat NULL
- Cardinalitatea maximă:
  - Câte rețete poate gestiona un angajat? => Multe
  - Câți angajați pot gestiona aceeași rețetă? => 1

### 3.2.2.3. *PACIENTI – RETETE*

**Relația:** *PACIENTI* dețin *RETETE*

**Cardinalități:**

- Cardinalitatea minimă:
  - Câte rețete trebuie să fie deținute de un pacient? => 0
    - pot exista pacienți pentru care să nu fie înregistrate rețete în baza de date
  - Câți pacienți trebuie să dețină aceeași rețetă? => 1
    - o rețetă poate fi deținută de un singur pacient
- Cardinalitatea maximă:
  - Câte rețete poate deține un pacient? => Multe
  - Câți pacienți pot deține aceeași rețetă? => 1

### 3.2.2.4. *MEDICI\_PRESCRIPTORI – RETETE*

**Relația:** *MEDICI\_PRESCRIPTORI* eliberează *RETETE*

**Cardinalități:**

- Cardinalitatea minimă:
  - Câte rețete trebuie să elibereze un medic prescriptor? => 0
    - pot exista medici care nu au eliberat nicio rețetă
  - Câți medici prescriptori trebuie să elibereze aceeași rețetă? => 1
    - o rețetă poate fi eliberată de un singur medic prescriptor
- Cardinalitatea maximă:
  - Câte rețete poate să elibereze un medic prescriptor? => Multe
  - Câți medici prescriptori pot să elibereze aceeași rețetă? => 1

### 3.2.2.5. *DIAGNOSTICE – RETETE*

**Relația:** *DIAGNOSTICE* sunt generate de *RETETE*

**Cardinalități:**

- Cardinalitatea minimă:
  - Câte rețete trebuie să fie generate de un diagnostic? => 0
    - pot exista diagnostice pentru care să nu fie înregistrate rețete în baza de date
  - Câte diagnostice trebuie să genereze o rețetă? => 1
    - o rețetă poate fi generată de un singur diagnostic
- Cardinalitatea maximă:
  - Câte rețete pot fi generate de același diagnostic? => Multe
  - Câte diagnostice poate genera o rețetă? => 1

### 3.2.2.6. *RETETE – MEDICAMENTE*

**Relația:** *RETETE* includ *MEDICAMENTE*

**Cardinalități:**

- Cardinalitatea minimă:
  - Câte medicamente trebuie să fie incluse pe o rețetă? => 1
    - o rețetă trebuie să includă cel puțin un medicament
  - Câte rețete trebuie să includă un anumit medicament? => 0
    - pot exista medicamente care să nu fie incluse pe nicio rețetă
- Cardinalitatea maximă:
  - Câte medicamente pot să fie incluse pe o rețetă? => Multe
  - Câte rețete pot să includă același medicament? => Multe

### 3.2.2.7. *MEDICAMENTE – PRODUCATORI*

**Relația:** *MEDICAMENTE* sunt fabricate de *PRODUCATORI*

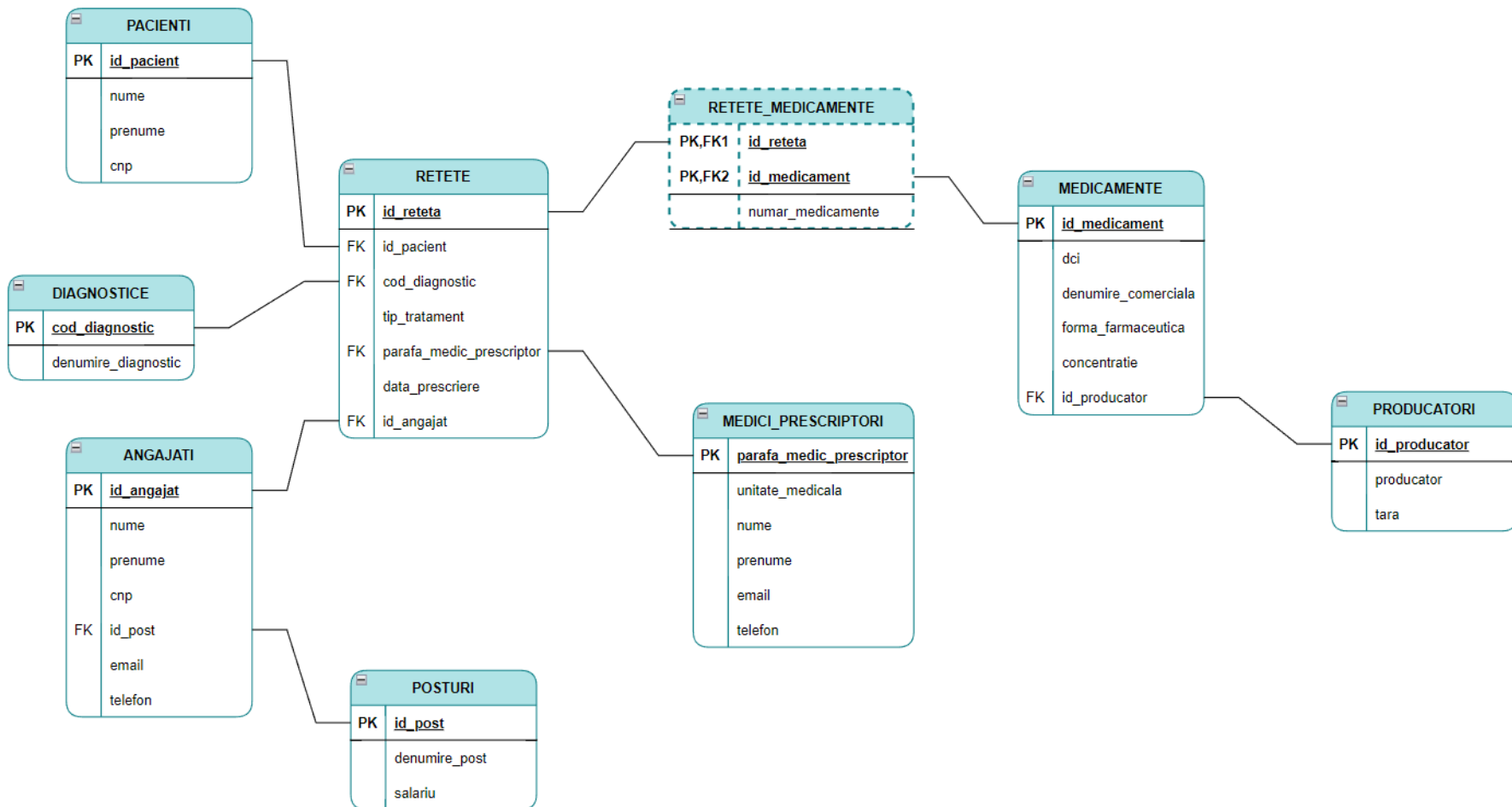
**Cardinalități:**

- Cardinalitatea minimă:
  - Câți producători trebuie să fabrice un medicament? => 0
    - în cazul în care un producător este șters, medicamentul rămâne în baza de date, dar cu atributul *producator* setat NULL
  - Câte medicamente trebuie să fie fabricate de un producător? => 0
    - pot exista producători care nu au fabricat niciun medicament momentan
- Cardinalitatea maximă:
  - Câți producători pot fabrica același medicament? => 1
  - Câte medicamente pot fi fabricate de un producător? => Multe



## 4. Diagrama conceptuală

### 4.1. Reprezentarea diagramei



## 4.2. Descrierea constrângerilor de integritate

### 4.2.1. Tabelul POSTURI

- **Constrângeri de tip PRIMARY KEY**
  - Atributul *id\_post*, a cărui constrângere va avea denumirea POST\_id\_post\_PK
- **Constrângeri de tip NOT NULL**
  - Atributul *denumire\_post*
    - Fiecare post trebuie să aibă o denumire
  - Atributul *salariu*
    - Fiecare post trebuie să aibă un salariu
- **Constrângeri de tip UNIQUE**
  - Atributul *denumire\_post*, a cărui constrângere se va numi POST\_den\_post\_UK
    - Nu pot exista mai multe posturi cu aceeași denumire
- **Constrângeri de tip CHECK**
  - Atributul *salariu*, a cărui constrângere se va numi POST\_salariu\_C
    - Se va verifica dacă salariul este un număr pozitiv, mai mare decât minimul pe economie, stabilit prin HG nr. 1071/2021 (*salariu* > 1525)

### 4.2.2. Tabelul ANGAJATI

- **Constrângeri de tip PRIMARY KEY**
  - Atributul *id\_angajat*, a cărui constrângere va avea denumirea ANG\_id\_ang\_PK
- **Constrângeri de tip FOREIGN KEY**
  - Atributul *id\_post*, a cărui constrângere va avea denumirea ANG\_id\_post\_FK
    - Referă tabelul POSTURI, coloana *id\_post*
- **Constrângeri de tip NOT NULL**
  - Atributul *nume*
    - Fiecare angajat trebuie să aibă un nume
  - Atributul *prenume*
    - Fiecare angajat trebuie să aibă un prenume
  - Atributul *cnp*
    - Fiecare angajat trebuie să aibă un CNP

- Atributul *email*
  - Fiecare angajat trebuie să aibă un email
- Atributul *telefon*
  - Fiecare angajat trebuie să aibă un număr de telefon
- **Constrângeri de tip UNIQUE**
- Atributul *cnp*, a cărui constrângere se va numi ANG\_cnp\_UK
  - Nu pot exista mai mulți angajați cu același CNP
- Atributul *email*, a cărui constrângere se va numi ANG\_email\_UK
  - Nu pot exista mai mulți angajați cu aceeași adresă de email
- Atributul *telefon*, a cărui constrângere se va numi ANG\_telefon\_UK
  - Nu pot exista mai mulți angajați cu același număr de telefon
- **Constrângeri de tip CHECK**
- Atributul *cnp*, a cărui constrângere se va numi ANG\_cnp\_C
  - Se va verifica dacă CNP-ul este format din 13 cifre
- Atributul *telefon*, a cărui constrângere se va numi ANG\_telefon\_C
  - Se va verifica dacă numărul de telefon este format din 10 cifre

#### 4.2.3. Tabelul PACIENTI

- **Constrângeri de tip PRIMARY KEY**
- Atributul *id\_pacient*, a cărui constrângere va avea denumirea PAC\_id\_pac\_PK
- **Constrângeri de tip NOT NULL**
- Atributul *nume*
  - Fiecare pacient trebuie să aibă un nume
- Atributul *prenume*
  - Fiecare pacient trebuie să aibă un prenume
- Atributul *cnp*
  - Fiecare pacient trebuie să aibă un CNP
- **Constrângeri de tip UNIQUE**
- Atributul *cnp*, a cărui constrângere se va numi PAC\_cnp\_UK
  - Nu pot exista mai mulți pacienți cu același CNP

- **Constrângeri de tip CHECK**
- Atributul *cnp*, a cărui constrângere se va numi PAC\_cnp\_C
  - Se va verifica dacă CNP-ul este format din 13 cifre

#### 4.2.4. Tabelul MEDICI\_PRESCRIPTORI

- **Constrângeri de tip PRIMARY KEY**
- Atributul *parafa\_medic\_prescriptor*, a cărui constrângere va avea denumirea MEDP\_par\_medp\_PK
- **Constrângeri de tip NOT NULL**
- Atributul *unitate\_medicala*
  - Fiecare medic prescriptor trebuie să aparțină unei unități medicale
- Atributul *nume*
  - Fiecare medic prescriptor trebuie să aibă un nume
- Atributul *prenume*
  - Fiecare medic prescriptor trebuie să aibă un prenume
- Atributul *email*
  - Fiecare medic prescriptor trebuie să aibă un email
- Atributul *telefon*
  - Fiecare medic prescriptor trebuie să aibă un număr de telefon
- **Constrângeri de tip UNIQUE**
- Atributul *email*, a cărui constrângere se va numi MEDP\_email\_UK
  - Nu pot exista mai mulți medici prescriptori cu aceeași adresă de email
- Atributul *telefon*, a cărui constrângere se va numi MEDP\_telefon\_UK
  - Nu pot exista mai mulți medici prescriptori cu același număr de telefon
- **Constrângeri de tip CHECK**
- Atributul *telefon*, a cărui constrângere se va numi MEDP\_telefon\_C
  - Se va verifica dacă numărul de telefon este format din 10 cifre

#### 4.2.5. Tabelul DIAGNOSTICE

- **Constrângeri de tip PRIMARY KEY**
  - Atributul *cod\_diagnostic*, cu constrângerea DIAG\_cod\_diag\_PK
- **Constrângeri de tip NOT NULL**
  - Atributul *denumire\_diagnostic*
    - Fiecare diagnostic are o denumire
- **Constrângeri de tip UNIQUE**
  - Atributul *denumire\_diagnostic*, cu constrângerea DIAG\_den\_diag\_UK
    - Nu pot exista mai multe diagnostice cu aceeași denumire

#### 4.2.6. Tabelul RETETE

- **Constrângeri de tip PRIMARY KEY**
  - Atributul *id\_reteta*, a cărei constrângere va avea denumirea RET\_id\_ret\_PK
- **Constrângeri de tip FOREIGN KEY**
  - Atributul *id\_pacient*, cu constrângerea RET\_id\_pacient\_FK
    - Referă tabelul PACIENTI, coloana *id\_pacient*
  - Atributul *cod\_diagnostic*, a cărei constrângere se va numi RET\_cod\_diag\_FK
    - Referă tabelul DIAGNOSTICE, coloana *cod\_diagnostic*
  - Atributul *parafa\_medic\_prescriptor*, cu constrângerea RET\_par\_medp\_FK
    - Referă tabelul MEDICI\_PRESCRIPTORI, coloana *parafa\_medic\_prescriptor*
  - Atributul *id\_angajat*, a cărei constrângere se va numi RET\_id\_ang\_FK
    - Referă tabelul ANGAJATI, coloana *id\_angajat*
- **Constrângeri de tip NOT NULL**
  - Atributul *id\_pacient*
    - Fiecare rețetă trebuie să fie deținută de un pacient
  - Atributul *cod\_diagnostic*
    - Fiecare rețetă trebuie să aibă codul diagnosticului
  - Atributul *tip\_tratament*
    - Fiecare rețetă trebuie să fie de un anumit tip (acut, subacut sau cronic)

- Atributul *parafa\_medic\_prescriptor*
  - Fiecare rețetă trebuie să fie eliberată de un medic prescriptor
- **Constrângeri de tip CHECK**
- Atributul *tip\_tratament*, a cărui constrângere se va numi RET\_tip\_trat\_C
  - Se va verifica dacă tipul tratamentului este acut, subacut sau cronic  
(tip\_tratament='acut' OR tip\_tratament='subacut' OR tip\_tratament='cronic')

#### 4.2.7. Tabelul RETETE\_MEDICAMENTE

- **Constrângeri de tip PRIMARY KEY**
- Atributele *id\_reteta* și *id\_medicament*, cu constrângerea RMED\_PK
- **Constrângeri de tip FOREIGN KEY**
- Atributul *id\_reteta*, cu constrângerea RMED\_id\_ret\_FK
  - Referă tabelul RETETE, coloana *id\_reteta*
- Atributul *id\_medicament*, cu constrângerea RMED\_id\_med\_FK
  - Referă tabelul MEDICAMENTE, coloana *id\_medicament*
- **Constrângeri de tip NOT NULL**
- Atributul *numar\_medicamente*
  - Este necesar să se precizeze numărul de medicamente de același tip de pe fiecare rețetă

#### 4.2.8. Tabelul MEDICAMENTE

- **Constrângeri de tip PRIMARY KEY**
- Atributul *id\_medicament*, cu constrângerea MED\_id\_med\_PK
- **Constrângeri de tip FOREIGN KEY**
- Atributul *id\_producator*, cu constrângerea MED\_id\_prod\_FK
  - Referă tabelul PRODUCATORI, coloana *id\_producator*
- **Constrângeri de tip NOT NULL**
- Atributul *dci*
  - Fiecare medicament trebuie să aibă un DCI

- Atributul *denumire\_comerciala*
  - Fiecare medicament trebuie să aibă o denumire comercială
- Atributul *forma\_farmaceutica*
  - Fiecare medicament trebuie să aibă o formă farmaceutică
- Atributul *concentratie*
  - Fiecare medicament trebuie să aibă o anumită concentrație

#### 4.2.9. Tabelul PRODUCATORI

- **Constrângeri de tip PRIMARY KEY**
- Atributul *id\_producator*, cu constrângerea PROD\_id\_prod\_PK
- **Constrângeri de tip NOT NULL**
- Atributul *producator*
  - Fiecare producător trebuie să aibă un nume
- Atributul *tara*
  - Fiecare producător trebuie să aibă sediul într-o anumită țară
- **Constrângeri de tip UNIQUE**
- Atributul *producator*, a cărei constrângere se va numi PROD\_prod\_UK
  - Nu pot exista mai mulți producători cu același nume

### 4.3. Schemele relaționale

#### 4.3.1. Schemele relaționale

Schemele relaționale ale diagramei conceptuale atașate sunt următoarele:

- **POSTURI**(*id\_post#*, *denumire\_post*, *salariu*)
- **ANGAJATI**(*id\_angajat#*, *nume*, *prenume*, *cnp*, *id\_post(FK)*, *email*, *telefon*)
- **PACIENTI**(*id\_pacient#*, *nume*, *prenume*, *cnp*)
- **MEDICI\_PRESCRIPTORI**(*parafa\_medic\_prescriptor#*, *unitate\_medicala*, *nume*, *prenume*, *email*, *telefon*)
- **DIAGNOSTICE**(*cod\_diagnostic#*, *denumire\_diagnostic*)
- **RETETE**(*id\_reteta#*, *id\_pacient(FK)*, *cod\_diagnostic(FK)*, *tip\_tratament*, *parafa\_medic\_prescriptor(FK)*, *data\_prescriere*, *id\_angajat(FK)*)
- **RETETE\_MEDICAMENTE**(*id\_reteta#(FK)*, *id\_medicament#(FK)*, *numar\_medicamente*)
- **MEDICAMENTE**(*id\_medicament#*, *dci*, *denumire\_comerciala*, *forma\_farmaceutica*, *concentratie*, *id\_producator(FK)*)
- **PRODUCATORI**(*id\_producator#*, *producator*, *tara*)



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

Constrângerile **ON DELETE**, corespunzătoare fiecărei constrângeri de tip **FOREIGN KEY**, sunt următoarele:

- **ANG\_id\_post\_FK, ON DELETE SET NULL**
  - În cazul în care este șters un post din baza de date, toți angajații care ocupau postul respectiv vor rămâne în farmacie, dar nu vor mai avea niciun post (atributul *id\_post* va fi setat NULL)
- **RET\_id\_pacient\_FK, ON DELETE CASCADE**
  - În momentul în care este șters un pacient, toate rețetele pacientului respectiv vor fi șterse
- **RET\_cod\_diag\_FK, ON DELETE CASCADE**
  - În momentul în care este șters un diagnostic, toate rețetele cu acel diagnostic vor fi șterse
- **RET\_par\_medp\_FK, ON DELETE CASCADE**
  - În momentul în care este șters un medic prescriptor, toate rețetele prescrise de acesta vor fi șterse
- **RET\_id\_ang\_FK, ON DELETE SET NULL**
  - În cazul în care este șters un angajat, toate rețetele gestionate de acesta vor rămâne înregistrate în baza de date, dar atributul *id\_angajat* va fi setat NULL
- **RMED\_id\_ret\_FK, ON DELETE CASCADE**
  - În momentul în care este ștersă o rețetă, medicamentele asociate rețetei respective vor fi șterse
- **RMED\_id\_med\_FK, ON DELETE CASCADE**
  - În cazul în care un medicament este șters, rețeta va continua doar cu restul medicamentelor
- **MED\_id\_prod\_FK, ON DELETE SET NULL**
  - În momentul în care este șters un producător, medicamentele vor rămâne în baza de date, dar cu atributul *id\_producator* setat NULL

## 5. Implementarea într-un SGBD

### 5.1. Introducere

Pentru implementarea bazei de date, am folosit MySQL. Scriptul este salvat cu numele **264\_Dinoiu\_Nicoleta\_Anastasia.sql** și conține, de asemenea, o comandă DROP, care va trebui înlăturată în momentul în care se rulează scriptul pentru prima dată. Întrucât tabelele nu există în acel moment, programul va afișa o eroare.

### 5.2. Crearea tabelelor și a constrângerilor

#### 5.2.1. Crearea tabelului POSTURI

```
CREATE TABLE POSTURI (  
    id_post DECIMAL(5, 0),  
        CONSTRAINT POST_id_post_PK PRIMARY KEY (id_post),  
    denumire_post VARCHAR(20) NOT NULL,  
        CONSTRAINT POST_den_post_UK UNIQUE (denumire_post),  
    salariu DECIMAL(4, 0) NOT NULL,  
        CONSTRAINT POST_salariu_C CHECK (salariu > 1525)  
);
```

#### 5.2.2. Crearea tabelului ANGAJATI

```
CREATE TABLE ANGAJATI (  
    id_angajat DECIMAL(5, 0),  
        CONSTRAINT ANG_id_ang_PK PRIMARY KEY (id_angajat),  
    nume VARCHAR(20) NOT NULL,  
    prenume VARCHAR(30) NOT NULL,  
    cnp DECIMAL(13, 0) NOT NULL,  
        CONSTRAINT ANG_cnp_UK UNIQUE (cnp),
```

```

        CONSTRAINT ANG_cnp_C CHECK (LENGTH(cnp)=13),

id_post DECIMAL(5, 0),

        CONSTRAINT ANG_id_post_FK FOREIGN KEY (id_post) REFERENCES
POSTURI (id_post) ON DELETE SET NULL,

email VARCHAR(30) NOT NULL,

        CONSTRAINT ANG_email_UK UNIQUE (email),

telefon VARCHAR(10) NOT NULL,

        CONSTRAINT ANG_telefon_UK UNIQUE (telefon),

        CONSTRAINT ANG_telefon_C CHECK (LENGTH(telefon)=10)

);

```

### 5.2.3. Crearea tabelului PACIENTI

```

CREATE TABLE PACIENTI (

    id_pacient DECIMAL(5, 0),

        CONSTRAINT PAC_id_pac_PK PRIMARY KEY (id_pacient),

    nume VARCHAR(20) NOT NULL,

    prenume VARCHAR(30) NOT NULL,

    cnp DECIMAL(13, 0) NOT NULL,

        CONSTRAINT PAC_cnp_UK UNIQUE (cnp),

        CONSTRAINT PAC_cnp_C CHECK (LENGTH(cnp)=13)

);

```

#### 5.2.4. Crearea tabelului MEDICI\_PRESCRIPTORI

```
CREATE TABLE MEDICI_PRESCRIPTORI (  
    parafa_medic_prescriptor DECIMAL(5, 0),  
    CONSTRAINT MEDP_par_medp_PK PRIMARY KEY  
    (parafa_medic_prescriptor),  
    unitate_medicala VARCHAR(50) NOT NULL,  
    nume VARCHAR(20) NOT NULL,  
    prenume VARCHAR(30) NOT NULL,  
    email VARCHAR(30) NOT NULL,  
    CONSTRAINT MEDP_email_UK UNIQUE (email),  
    telefon VARCHAR(10) NOT NULL,  
    CONSTRAINT MEDP_telefon_UK UNIQUE (telefon),  
    CONSTRAINT MEDP_telefon_C CHECK (LENGTH(telefon)=10)  
);
```

#### 5.2.5. Crearea tabelului DIAGNOSTICE

```
CREATE TABLE DIAGNOSTICE (  
    cod_diagnostic DECIMAL(5, 0),  
    CONSTRAINT DIAG_cod_diag_PK PRIMARY KEY(cod_diagnostic),  
    denumire_diagnostic VARCHAR(100) NOT NULL,  
    CONSTRAINT DIAG_den_diag_UK UNIQUE (denumire_diagnostic)  
);
```

### 5.2.6. Crearea tabelului RETETE

```
CREATE TABLE RETETE (  
    id_reteta DECIMAL(5, 0),  
        CONSTRAINT RET_id_ret_PK PRIMARY KEY(id_reteta),  
    id_pacient DECIMAL(5, 0) NOT NULL,  
        CONSTRAINT RET_id_pacient_FK FOREIGN KEY(id_pacient)  
REFERENCES PACIENTI(id_pacient) ON DELETE CASCADE,  
    cod_diagnostic DECIMAL(5, 0) NOT NULL,  
        CONSTRAINT RET_cod_diag_FK FOREIGN KEY(cod_diagnostic)  
REFERENCES DIAGNOSTICE(cod_diagnostic) ON DELETE CASCADE,  
    tip_tratament VARCHAR(7) NOT NULL  
        CONSTRAINT RET_tip_trat_C CHECK (tip_tratament='acut' OR  
tip_tratament='subacut' OR tip_tratament='cronic'),  
    parafa_medic_prescriptor DECIMAL(5, 0) NOT NULL,  
        CONSTRAINT RET_par_medp_FK FOREIGN  
KEY(parafa_medic_prescriptor) REFERENCES  
MEDICI_PRESCRIPTORI(parafa_medic_prescriptor) ON DELETE CASCADE,  
    data_prescriere DATE,  
    id_angajat DECIMAL(5, 0),  
        CONSTRAINT RET_id_ang_FK FOREIGN KEY(id_angajat) REFERENCES  
ANGAJATI(id_angajat) ON DELETE SET NULL  
);
```

### 5.2.7. Crearea tabelului PRODUCATORI

```
CREATE TABLE PRODUCATORI (  
    id_producator DECIMAL(5, 0),  
        CONSTRAINT PROD_id_prod_PK PRIMARY KEY(id_producator),  
    producator VARCHAR(50) NOT NULL,  
        CONSTRAINT PROD_prod_UK UNIQUE(producator),  
    tara VARCHAR(20) NOT NULL  
);
```

### 5.2.8. Crearea tabelului MEDICAMENTE

```
CREATE TABLE MEDICAMENTE (  
    id_medicament DECIMAL(5, 0),  
        CONSTRAINT MED_id_med_PK PRIMARY KEY(id_medicament),  
    dci VARCHAR(50) NOT NULL,  
    denumire_comerciala VARCHAR(30) NOT NULL,  
    forma_farmaceutica VARCHAR(50) NOT NULL,  
    concentratie VARCHAR(30) NOT NULL,  
    id_producator DECIMAL(5, 0),  
        CONSTRAINT MED_id_prod_FK FOREIGN KEY(id_producator)  
REFERENCES PRODUCATORI(id_producator) ON DELETE SET NULL  
);
```

### 5.2.9. Crearea tabelului RETETE\_MEDICAMENTE

```
CREATE TABLE RETETE_MEDICAMENTE (  
    id_reteta DECIMAL(5, 0),  
    CONSTRAINT RMED_id_ret_FK FOREIGN KEY (id_reteta) REFERENCES  
    RETETE(id_reteta) ON DELETE CASCADE,  
    id_medicament DECIMAL(5, 0),  
    CONSTRAINT RMED_id_med_FK FOREIGN KEY (id_medicament)  
    REFERENCES MEDICAMENTE(id_medicament) ON DELETE CASCADE,  
    numar_medicamente DECIMAL(5, 0) NOT NULL,  
    CONSTRAINT RMED_PK PRIMARY KEY (id_reteta, id_medicament)  
);
```

## 5.3. Introducerea datelor

### 5.3.1. Introducerea datelor în tabelul POSTURI

```
INSERT INTO POSTURI (id_post, denumire_post, salariu) VALUES (1,  
'Farmacist sef', 5000);  
  
INSERT INTO POSTURI (id_post, denumire_post, salariu) VALUES (2,  
'Farmacist', 3500);  
  
INSERT INTO POSTURI (id_post, denumire_post, salariu) VALUES (3,  
'Asistent', 2300);  
  
INSERT INTO POSTURI (id_post, denumire_post, salariu) VALUES (4,  
'Ingrijitor', 1800);  
  
INSERT INTO POSTURI (id_post, denumire_post, salariu) VALUES (5,  
'Agent de paza', 1800);
```

### 5.3.2. Introducerea datelor în tabelul ANGAJATI

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (1, 'Cantemir', 'Ioana', '2900112416125', 1, 'icantemir@gmail.com', '0767781869');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (2, 'Dumitrescu', 'Stefan', '2900908416467', 2, 'sdumitrescu@gmail.com', '0768391749');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (3, 'Popescu', 'Lucia', '2880902419651', 2, 'lpopescu@gmail.com', '0760945166');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (4, 'Stoica', 'Mariana', '2910904418185', 2, 'mstoica@gmail.com', '0769180843');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (5, 'Munteanu', 'Andrada', '2910904419312', 3, 'amunteanu@gmail.com', '0725473119');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (6, 'Serban', 'Laurentiu', '2910905415940', 3, 'lserban@gmail.com', '0721427665');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (7, 'Toma', 'Andreea', '2910905415174', 3, 'atoma@gmail.com', '0723929887');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (8, 'Lazar', 'Florentina', '2910905418997', 3, 'flazar@gmail.com', '0724969576');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (9, 'Florea', 'Liliana', '2770422416169', 3, 'lflorea@gmail.com', '0721443546');
```



```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (10, 'Tudor', 'Cornelia', '2770422415951', 4, 'ctudor@gmail.com', '0728844502');
```

```
INSERT INTO ANGAJATI (id_angajat, nume, prenume, cnp, id_post, email, telefon) VALUES (11, 'Dragomir', 'Alexandru', '2810409418360', 5, 'adragomir@gmail.com', '0729497262');
```

### 5.3.3. Introducerea datelor în tabelul PACIENTI

```
INSERT INTO PACIENTI (id_pacient, nume, prenume, cnp) VALUES (1, 'Vlad', 'Marina', '2910401438968');
```

```
INSERT INTO PACIENTI (id_pacient, nume, prenume, cnp) VALUES (2, 'Iordache', 'Larisa', '2910402096305');
```

```
INSERT INTO PACIENTI (id_pacient, nume, prenume, cnp) VALUES (3, 'Ion', 'Ciprian', '1930402409522');
```

```
INSERT INTO PACIENTI (id_pacient, nume, prenume, cnp) VALUES (4, 'Tanase', 'Laura', '2910403408726');
```

```
INSERT INTO PACIENTI (id_pacient, nume, prenume, cnp) VALUES (5, 'Craciun', 'Larisa', '2910404408181');
```

### 5.3.4. Introducerea datelor în tabelul MEDICI\_PRESCRIPTORI

```
INSERT INTO MEDICI_PRESCRIPTORI (parafa_medic_prescriptor, unitate_medicala, nume, prenume, email, telefon) VALUES (1, 'CMI DR. Suciu Georgeta', 'Suciu', 'Georgeta', 'gsuciu@gmail.com', '0712088686');
```

```
INSERT INTO MEDICI_PRESCRIPTORI (parafa_medic_prescriptor, unitate_medicala, nume, prenume, email, telefon) VALUES (2, 'CMI DR. Marin Eugen', 'Marin', 'Eugen', 'emarin@gmail.com', '0724757789');
```

```
INSERT INTO MEDICI_PRESCRIPTORI (parafa_medic_prescriptor, unitate_medicala, nume, prenume, email, telefon) VALUES (3, 'CMI DR.
```

```
Ionescu Viorel', 'Ionescu', 'Viorel', 'vionescu@gmail.com',  
'0718186848');
```

```
INSERT INTO MEDICI_PRESCRIPTORI (parafa_medic_prescriptor,  
unitate_medicala, nume, prenume, email, telefon) VALUES (4, 'CMI DR.  
Munteanu Elena', 'Munteanu', 'Elena', 'emunteanu@gmail.com',  
'0759202683');
```

```
INSERT INTO MEDICI_PRESCRIPTORI (parafa_medic_prescriptor,  
unitate_medicala, nume, prenume, email, telefon) VALUES (5, 'CMI DR.  
Rizea Valeria', 'Rizea', 'Valeria', 'vrizea@gmail.com', '0716042906');
```

#### 5.3.5. Introducerea datelor în tabelul DIAGNOSTICE

```
INSERT INTO DIAGNOSTICE (cod_diagnostic, denumire_diagnostic) VALUES  
(1, 'Gastrita si duodenita');
```

```
INSERT INTO DIAGNOSTICE (cod_diagnostic, denumire_diagnostic) VALUES  
(2, 'Dispepsia');
```

```
INSERT INTO DIAGNOSTICE (cod_diagnostic, denumire_diagnostic) VALUES  
(3, 'Hipertensiunea esentiala (primara)');
```

```
INSERT INTO DIAGNOSTICE (cod_diagnostic, denumire_diagnostic) VALUES  
(4, 'Fibroza si ciroza ficatului');
```

```
INSERT INTO DIAGNOSTICE (cod_diagnostic, denumire_diagnostic) VALUES  
(5, 'Vene varicoase ale membrelor inferioare');
```

#### 5.3.6. Introducerea datelor în tabelul RETETE

```
INSERT INTO RETETE (id_reteta, id_pacient, cod_diagnostic,  
tip_tratament, parafa_medic_prescriptor, data_prescriere, id_angajat)  
VALUES (1, 1, 1, 'subacut', 1, '2022-01-03', 1);
```

```
INSERT INTO RETETE (id_reteta, id_pacient, cod_diagnostic,  
tip_tratament, parafa_medic_prescriptor, data_prescriere, id_angajat)  
VALUES (2, 1, 2, 'subacut', 1, '2022-01-03', 1);
```

```
INSERT INTO RETETE (id_reteta, id_pacient, cod_diagnostic,  
tip_tratament, parafa_medic_prescriptor, data_prescriere, id_angajat)  
VALUES (3, 2, 3, 'cronic', 2, '2022-01-03', 2);
```

```
INSERT INTO RETETE (id_reteta, id_pacient, cod_diagnostic,  
tip_tratament, parafa_medic_prescriptor, data_prescriere, id_angajat)  
VALUES (4, 3, 4, 'cronic', 3, '2022-01-04', 3);
```

```
INSERT INTO RETETE (id_reteta, id_pacient, cod_diagnostic,  
tip_tratament, parafa_medic_prescriptor, data_prescriere, id_angajat)  
VALUES (5, 4, 5, 'cronic', 4, '2022-01-05', 2);
```

#### **5.3.7. Introducerea datelor în tabelul PRODUCATORI**

```
INSERT INTO PRODUCATORI (id_producator, producator, tara) VALUES (1,  
'ASTRAZENECA AB', 'Suedia');
```

```
INSERT INTO PRODUCATORI (id_producator, producator, tara) VALUES (2,  
'TERAPIA S.A.', 'Romania');
```

```
INSERT INTO PRODUCATORI (id_producator, producator, tara) VALUES (3,  
'BERLIN - CHEMIE AG', 'Germania');
```

```
INSERT INTO PRODUCATORI (id_producator, producator, tara) VALUES (4,  
'LES LABORATOIRES SERVIER INDUSTRIE (LSI)', 'Franta');
```

```
INSERT INTO PRODUCATORI (id_producator, producator, tara) VALUES (5,  
'MEDIS INTERNATIONAL A.S', 'Republica Ceha');
```

```
INSERT INTO PRODUCATORI (id_producator, producator, tara) VALUES (6,  
'ALFASIGMA S.P.A.', 'Italia');
```

#### **5.3.8. Introducerea datelor în tabelul MEDICAMENTE**

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,  
forma_farmaceutica, concentratie, id_producator) VALUES (1,  
'ESOMEPRAZOLUM', 'NEXIUM', 'COMPR. GASTROREZISTENTE', '40mg', 1);
```

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,
forma_farmaceutica, concentratie, id_producator) VALUES (2,
'DOMPERIDONUM', 'MOTILIUM', 'COMPR. FILM.', '10mg', 2);
```

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,
forma_farmaceutica, concentratie, id_producator) VALUES (3,
'NEBIVOLOLUM', 'NEBILET', 'COMPR.', '5mg', 3);
```

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,
forma_farmaceutica, concentratie, id_producator) VALUES (4,
'COMBINATII (PERINDOPRILUM+INDAPAMIDUM)', 'NOLIPREL ARG', 'COMPR.
FILM.', '2,5mg/0,625mg', 4);
```

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,
forma_farmaceutica, concentratie, id_producator) VALUES (5,
'FUROSEMIDUM', 'FUROSEMID MEDREG', 'COMPR.', '40mg', 5);
```

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,
forma_farmaceutica, concentratie, id_producator) VALUES (6,
'ESOMEPRAZOLUM', 'ESOMEPRAZOL TERAPIA', 'CAPS. GASTROREZ.', '40mg',
2);
```

```
INSERT INTO MEDICAMENTE (id_medicament, dci, denumire_comerciala,
forma_farmaceutica, concentratie, id_producator) VALUES (7,
'SULODEXIDUM', 'VESSEL DUE F', 'CAPS. MOI', '250ULS', 6);
```

### **5.3.9. Introducerea datelor în tabelul RETETE\_MEDICAMENTE**

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,
numar_medicamente) VALUES (1, 1, 1);
```

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,
numar_medicamente) VALUES (2, 2, 1);
```

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,
numar_medicamente) VALUES (3, 3, 2);
```

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,  
numar_medicamente) VALUES (3, 4, 1);
```

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,  
numar_medicamente) VALUES (4, 5, 1);
```

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,  
numar_medicamente) VALUES (4, 6, 1);
```

```
INSERT INTO RETETE_MEDICAMENTE (id_reteta, id_medicament,  
numar_medicamente) VALUES (5, 7, 1);
```