

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

**PROIECT
BAZE DE DATE**

PROFESOR COORDONATOR:
VASILE SILVIU LAURENȚIU

STUDENT:
UDREA IULIA-MARIA

BUCUREȘTI

2022

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

**ADMINISTRAREA UNEI
CLINICI DENTARE**

PROFESOR COORDONATOR:
VASILE SILVIU LAURENȚIU

STUDENT:
UDREA IULIA-MARIA

BUCUREȘTI

2022

CUPRINS

1. Introducere.....	4
2. Regulile modelului	4
3. Diagrama entitate-relație.....	5
4. Descrierea entităților, atributelor și a cheilor.....	5
4.1.Tabelul SERVICII.....	5
4.2.Tabelul PACIENTI	5
4.3.Tabelul JOBURI	6
4.4.Tabelul ANGAJATI	6
4.5.Tabelul SALI	7
4.6.Tabelul FURNIZORI.....	7
4.7.Tabelul ECHIPAMENTE	7
5. Descrierea relațiilor și a cardinalităților	8
5.1.Relția SERVICII – PACIENTI	8
5.2.Relția ANGAJATI – SERVICII	8
5.3.Relția PACIENTI – ANGAJATI.....	9
5.4. Relția ANGAJATI – JOBURI	9
5.5. Relția ANGAJATI – SALI	9
5.6.Relția SALI – ECHIPAMENTE	10
5.7.Relția ECHIPAMENTE – FURNIZORI.....	10
6. Diagrama conceptuală.....	11
7. Descrierea constrângerilor de integritate	12
7.1. Tabelul SERVICII	12
7.2. Tabelul PACIENTI.....	13
7.3. Tabelul JOBURI.....	13
7.4. Tabelul ANGAJATI.....	14
7.5.Tabelul SALI	14
7.6.Tabelul FURNIZORI.....	15
7.7.Tabelul ECHIPAMENTE	15
7.8.Tabelul PROGRAMARI.....	16

7.9.Tabelul REPARTIZARE_SALI.....	16
7.10.Tabelul INVENTARE.....	17
7.11.Tabelul LIVRARI.....	17
8. Schemele relaționale.....	18
9. Constrângerile de ștergere	18
10.Crearea tabelelor.....	19
10.1.Tabelul SERVICII.....	19
10.2.Tabelul PACIENTI	19
10.3.Tabelul JOBURI	20
10.4.Tabelul ANGAJATI	20
10.5.Tabelul SALI	21
10.6.Tabelul FURNIZORI.....	21
10.7.Tabelul ECHIPAMENTE	21
10.8.Tabelul PROGRAMARI.....	22
10.9.Tabelul REPARTIZARE_SALI.....	22
10.10.Tabelul INVENTARE.....	23
10.11.Tabelul LIVRARI	23
11.Inserarea datelor în table.....	23
11.1.Tabelul SERVICII.....	24
11.2.Tabelul PACIENTI	24
11.3.Tabelul JOBURI	24
11.4.Tabelul ANGAJATI	25
11.5.Tabelul SALI	25
11.6.Tabelul FURNIZORI	25
11.7.Tabelul ECHIPAMENTE	26
11.8.Tabelul PROGRAMARI.....	26
11.9.Tabelul REPARTIZARE_SALI.....	26
11.10.Tabelul INVENTARE.....	27
11.11.Tabelul LIVRARI	27

1. Introducere

O clinică dentară reunește medici specialiști din diverse ramuri ale stomatologiei, astfel încât un pacient poate beneficia de servicii complete într-un singur loc. Pentru o bună organizare, toți pacienții trebuie să se înregistreze în clinică (să completeze un formular cu datele acestora) înainte de a solicita servicii. În realizarea serviciilor, angajații folosesc echipamente, instrumente și materiale, provenite de la mai mulți furnizori.

Administrarea unei clinici dentare presupune, în primul rând, evidența pacienților, a serviciilor pe care le solicită și a interacțiunii acestora cu angajații. Se monitorizează, de asemenea, furnizarea echipamentelor necesare și distribuția lor în interiorul clinicii.

În realizarea bazei de date, am considerat câteva aspecte care definesc clinica: un serviciu durează cel mult o oră, iar o programare se poate face doar la oră fixă. Clinica este deschisă zilnic, între orele 9-17, și se extinde pe două etaje și parter. Aceste informații vor limita introducerea înregistrărilor în baza de date. Implementarea acesteia a fost realizată folosind MySQL 8.0.27.

2. Regulile modelului

- Un serviciu poate fi solicitat de unul sau mai mulți pacienți.

Un pacient poate solicita unul sau mai multe servicii.

- Un serviciu poate fi prestat de unul sau mai mulți angajați.

Un angajat poate presta unul sau mai multe servicii.

- Un angajat poate trata unul sau mai mulți pacienți.

Un pacient poate fi tratat de unul sau mai mulți angajați.

- Un angajat poate să fie managerul unuia sau mai multor angajați.

Un angajat poate să fie subalternul unui singur angajat.

- Un angajat trebuie să aibă un singur job.

Un job poate să aparțină unuia sau mai multor angajați.

- Un angajat trebuie să lucreze în una sau mai multe săli.

O sală poate fi repartizată unuia sau mai multor angajați.

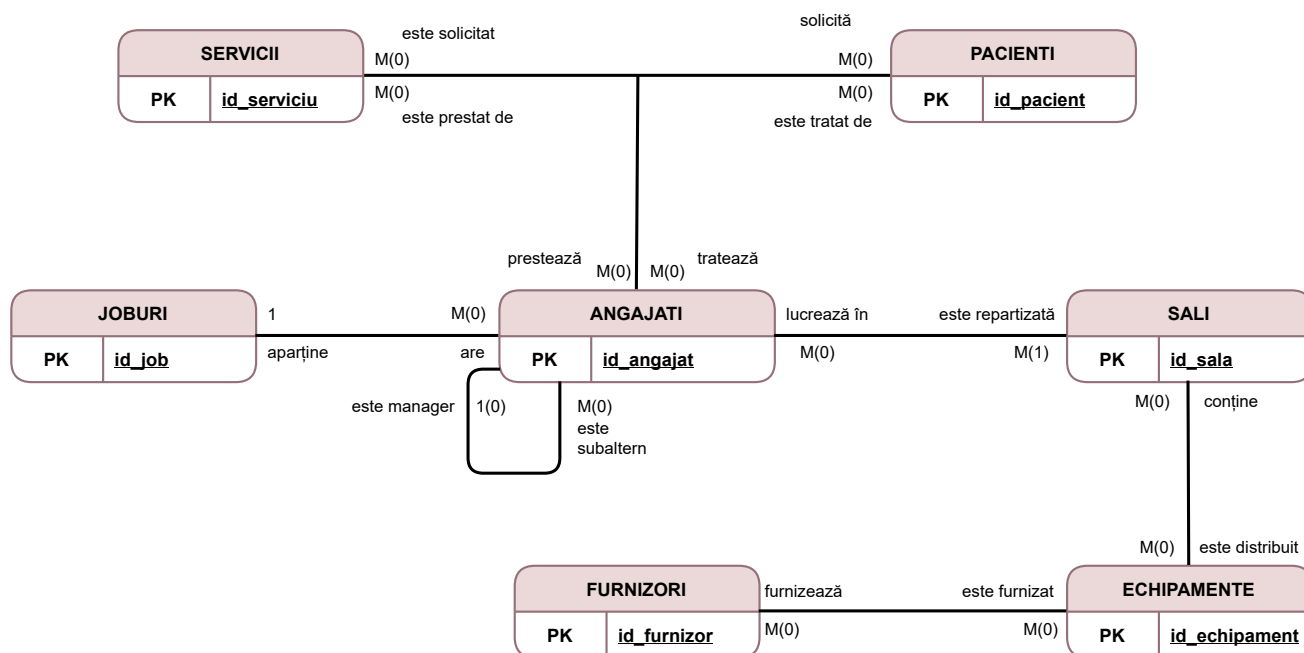
- Un echipament poate fi distribuit în unul sau mai multe săli.

O sală poate conține unul sau mai multe echipamente.

- Un echipament poate fi furnizat de unul sau mai mulți furnizori.

Un furnizor poate furniza unul sau mai multe echipamente.

3. Diagrama entitate-relație



4. Descrierea entităților, atributelor și a cheilor

Entitățile au attribute care le caracterizează, printre care și o cheie primară (PK). Anumite attribute sunt preluate din attributele altei entități (de exemplu, un angajat are atributul `id_job`, preluat de la entitatea `JOBURI`), acestea reprezentând cheile externe (FK). În diagrama conceptuală, entitățile se vor transforma în tabele, iar attributele în coloane. Cheia primară va identifica unic înregistrările din tabel, în timp ce cheile externe vor referi coloane din alte tabele.

4.1. Tabelul SERVICII

- conține informații despre serviciile ce pot fi oferite pacienților clinicii;

Cheie	Atribut	Tip de date	Descriere
PK	<code>id_serviciu</code>	integer	Identificatorul unic al serviciului
-	<code>denumire</code>	<code>varchar(50)</code>	Denumirea serviciului
-	<code>specializare</code>	<code>varchar(20)</code>	Specializarea din care aparține serviciul; Aceasta poate fi una din următoarele: <code>stom_gen</code> , <code>ortodontie</code> , <code>implantologie</code> , <code>estetica</code> , <code>endodontie</code> , <code>pedodontie</code> , <code>protetica</code> , <code>chirurgie</code> , <code>radiologie</code> , <code>profilaxie</code>
-	<code>pret</code>	double	Prețul serviciului
-	<code>durata</code>	integer	Durata serviciului, măsurată în minute

4.2. Tabelul PACIENTI

- conține informații despre persoanele care s-au înscris în cadrul clinicii (inclusiv despre cei care nu au beneficiat încă de un serviciu);

Cheie	Atribut	Tip de date	Descriere
PK	id_pacient	integer	Identificatorul unic al pacientului
-	nume	varchar(20)	Numele de familie al pacientului
-	prenume	varchar(20)	Prenumele pacientului
-	data_nasterii	date	Data nașterii pacientului
-	telefon	varchar(10)	Numărul de telefon al pacientului
-	email	varchar(40)	Adresa de email a pacientului

4.3. Tabelul JOBURI

- conține informații despre joburile pe care le pot deține angajații;

Cheie	Atribut	Tip de date	Descriere
PK	id_job	integer	Identificatorul unic al jobului
-	titlu	varchar(30)	Titlul jobului
-	salariu_min	integer	Salariul minim care poate fi oferit pentru job
-	salariu_max	integer	Salariul maxim care poate fi oferit pentru job
-	studii_necesare	varchar(20)	Nivelul minim de studii necesar pentru job

4.4. Tabelul ANGAJATI

- conține informații despre angajații clinicii;

Cheie	Atribut	Tip de date	Descriere
PK	id_angajat	integer	Identificatorul unic al angajatului
FK	id_job	integer	Identificatorul jobului pe care îl deține angajatul
FK	id_manager	integer	Identificatorul angajatului care este manager pentru angajatul introdus
-	nume	varchar(20)	Numele de familie al angajatului
-	prenume	varchar(20)	Prenumele angajatului
-	data_nasterii	date	Data nașterii angajatului
-	telefon	varchar(10)	Numărul de telefon al angajatului
-	email	varchar(40)	Adresa de email a angajatului
-	salariu	double	Salariul angajatului

4.5. Tabelul SALI

- conține informații despre sălile în care își desfășoară activitatea angajații;

Cheie	Atribut	Tip de date	Descriere
PK	id_sala	integer	Identificatorul unic al sălii
-	tip	varchar(10)	Tipul sălii; Poate fi unul din următoarele: cabinet, laborator, birou, recepție
-	suprafata	double	Suprafața încăperii, măsurată în m^2
-	etaj	integer	Etajul pe care se află sala; Se încadrează între 0 (parter) și 2 (clinica are două etaje)

4.6. Tabelul FURNIZORI

- conține informații despre furnizorii cu care poate colabora clinica;

Cheie	Atribut	Tip de date	Descriere
PK	id_furnizor	integer	Identificatorul unic al furnizorului
-	nume	varchar(30)	Numele furnizorului
-	telefon	varchar(10)	Numărul de telefon al furnizorului
-	email	varchar(30)	Adresa de email a furnizorului

4.7. Tabelul ECHIPAMENTE

- conține informații despre echipamentele, instrumentele și materialele generale de care este nevoie în clinică (nu sunt neapărat echipamente pe care clinica le deține; pot să fie echipamente pe care clinica și-a propus să le achiziționeze și care încă nu au fost livrate);

Cheie	Atribut	Tip de date	Descriere
PK	id_echipament	integer	Identificatorul unic al echipamentului
-	denumire	varchar(50)	Denumirea echipamentului
-	categorie	varchar(30)	Categoria din care face parte echipamentul
-	pret	double	Prețul echipamentului

5. Descrierea relațiilor și a cardinalităților

Relațiile reprezintă legături între entități, pentru fiecare relație fiind stabilite cardinalitatea minimă și cardinalitatea maximă. În diagrama conceptuală, relațiile se transformă fie în tabele asociative, fie în chei externe (și, eventual, pot să facă parte dintr-o cheie primară). Cardinalitățile se vor regăsi în constrângerile aplicate coloanelor care reprezintă chei externe.

5.1. Relația SERVICII – PACIENTI

De câți pacienți poate fi solicitat un serviciu? **Mulți**

De câți pacienți trebuie să fie solicitat un serviciu? **Zero**

ex: Clinica poate să ofere servicii pe care nu le-a solicitat nimeni.

⇒ Un serviciu **poate** fi solicitat de **unul sau mai mulți** pacienți.

Câte servicii poate să solicite un pacient? **Multe**

Câte servicii trebuie să solicite un pacient? **Zero**

ex: Un pacient se înscrie preventiv în clinică, fără să solicite neapărat un serviciu.

⇒ Un pacient **poate** să solicite **unul sau mai multe** servicii.

- Cardinalitate maximă: many-many (M:N)
- Cardinalitate minimă: zero-zero (0:0)

5.2. Relația ANGAJATI – SERVICII

Câte servicii poate să presteze un angajat? **Multe**

Câte servicii trebuie să presteze un angajat? **Zero**

ex: Putem vorbi despre un angajat nou care nu a prestat încă servicii, sau despre un angajat care nu prestează servicii pentru pacienți (precum un manager).

⇒ Un angajat **poate** să presteze **unul sau mai multe** servicii.

De câți angajați poate fi prestat un serviciu? **Mulți**

De câți angajați trebuie să fie prestat un serviciu? **Zero**

⇒ Un serviciu **poate** fi prestat de **unul sau mai mulți** angajați.

- Cardinalitate maximă: many-many (M:N)
- Cardinalitate minimă: zero-zero (0:0)

5.3. Relația PACIENTI – ANGAJATI

De câți angajați poate fi tratat un pacient? **Mulți**

De câți angajați trebuie să fie tratat un pacient? **Zero**

ex: Un pacient nu a solicitat niciun serviciu, deci nu a fost tratat de niciun angajat.

⇒ Un pacient **poate** fi tratat de **unul sau mai mulți** angajați.

Câți pacienți pot fi tratați de un anumit angajat? **Mulți**

Câți pacienți trebuie să fie tratați de un anumit angajat? **Zero**

ex: La fel ca la relația anterioară, poate fi vorba despre un angajat nou care nu a tratat încă pe nimeni, sau un angajat care nu tratează pacienți.

⇒ Un angajat **poate** să trateze **unul sau mai mulți** pacienți.

- Cardinalitate maximă: many-many (M:N)
- Cardinalitate minimă: zero-zero (0:0)

5.4. Relația ANGAJATI – JOBURI

Câte joburi poate să aibă un angajat? **Unul**

Câte joburi trebuie să aibă un angajat? **Unul**

⇒ Un angajat **trebuie** să aibă **un singur** job.

Câți angajați pot să dețină un anumit job? **Mulți**

Câți angajați trebuie să dețină un anumit job? **Zero**

ex: Pot exista joburi pentru care nu a fost încă angajată nicio persoană.

⇒ Un job **poate** să aparțină **unuia sau mai multor** angajați.

- Cardinalitate maximă: one-many (1:M)
- Cardinalitate minimă: one-zero (1:0)

5.5. Relația ANGAJATI – SALI

În câte săli poate să lucreze un angajat? **Multe**

În câte săli trebuie să lucreze un angajat? **Una**

⇒ Un angajat **trebuie** să lucreze în **una sau mai multe** săli.

Câți angajați pot să fie repartizați într-o anumită sală? **Mulți**

Câți angajați trebuie să fie repartizați într-o anumită sală? **Zero**

⇒ O sală **poate** fi repartizată **unuia sau mai multor** angajați.

- Cardinalitate maximă: many-many (M:N)
- Cardinalitate minimă: one-zero (1:0)

5.6. Relația SALI – ECHIPAMENTE

Câte echipamente poate să conțină o sală? **Multe**

Câte echipamente trebuie să conțină o sală? **Zero**

ex: O sală poate să nu fie momentan dată în folosință sau poate fi în renovare.

⇒ O sală **poate** să conțină **unul sau mai multe** echipamente.

În câte săli poate fi distribuit un echipament? **Mule**

În câte săli trebuie să fie distribuit un echipament? **Zero**

⇒ Un echipament **poate** să fie distribuit în **una sau mai multe** săli.

- Cardinalitate maximă: many-many (M:N)
- Cardinalitate minimă: zero-zero (0:0)

5.7. Relația ECHIPAMENTE – FURNIZORI

De câți furnizori poate fi furnizat un echipament? **Mulți**

ex: Echipamentele sunt generale (de ex., ”clește extracție”), deci un echipament poate proveni de la mai mulți furnizori.

De câți furnizori trebuie să fie furnizat un echipament? **Zero**

ex: Clinica are nevoie de un echipament care nu a fost furnizat.

⇒ Un echipament **poate** fi furnizat de **unul sau mai mulți** furnizori.

Câte echipamente pot fi furnizate de un furnizor? **Multe**

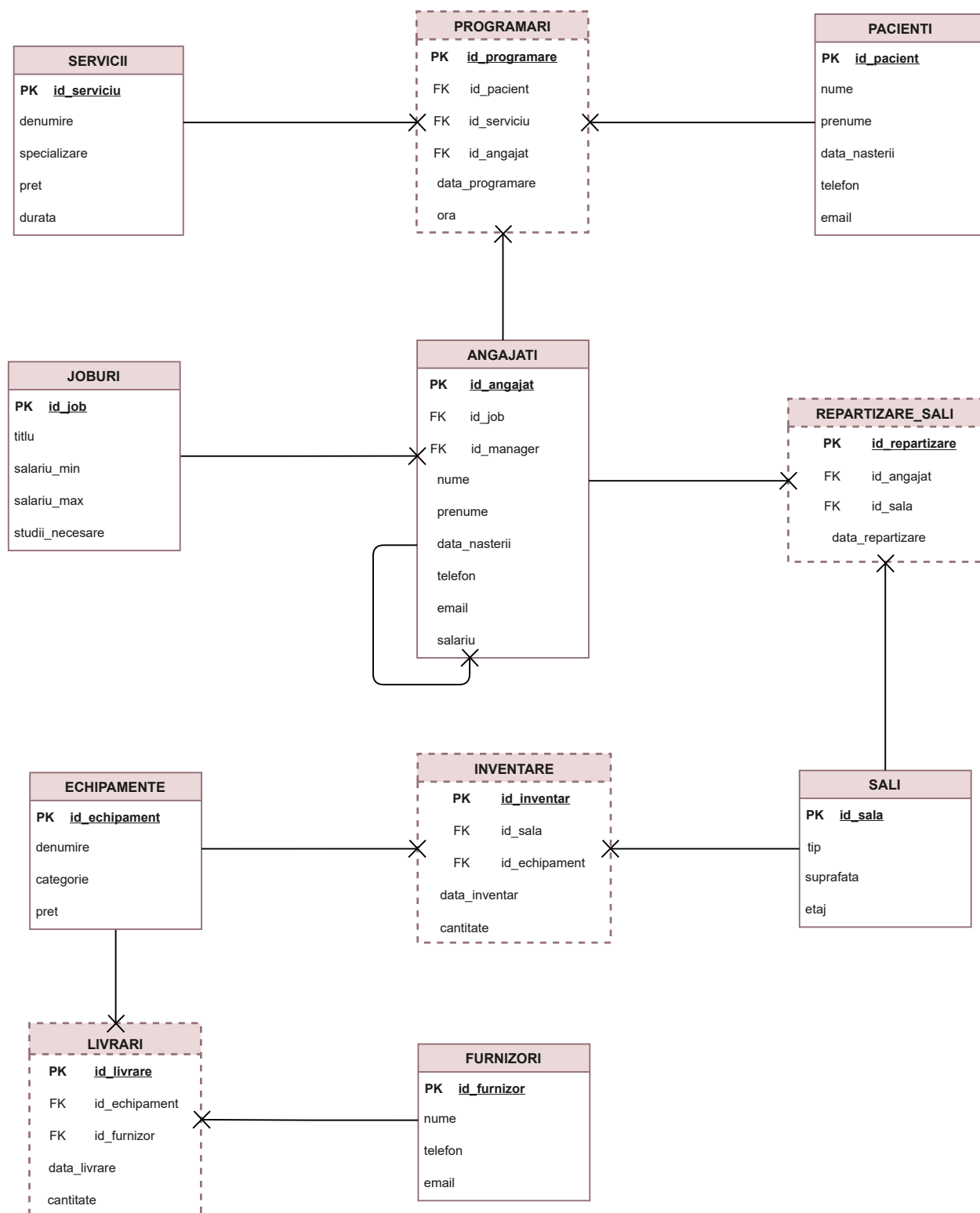
Câte echipamente trebuie să fie furnizate de un furnizor? **Zero**

ex: Furnizorul nu a livrat niciun echipament.

⇒ Un furnizor **poate** să furnizeze **unul sau mai multe** echipamente.

- Cardinalitate maximă: many-many (M:N)
- Cardinalitate minimă: zero-zero (0:0)

6. Diagrama conceptuală



7. Descrierea constrângerilor de integritate

Constrângerile de integritate limitează inserarea, actualizarea și ștergerea datelor din tabele. Cheia primară (PK) identifică unic înregistrările din tabel, iar cheile externe (FK) trebuie să refere coloane din alte tabele. Constrângerea NOT NULL se aplică unei coloane și impune ca valorile ei să fie diferite de null; constrângerile UNIQUE verifică unicitatea valorilor unei coloane sau a unei combinații de coloane în tabel, iar constrângerile CHECK permit realizarea unor verificări asupra valorilor dintr-o înregistrare.

Unele constrângeri aplicate cheilor externe reies din cardinalitățile relațiilor. De exemplu, cardinalitățile din relația ”Un angajat trebuie să aibă un singur job” impun ca id_job (cheie externă provenită din tabelul JOBURI) din tabelul angajați să fie diferită de *null*.

7.1. Tabelul SERVICII

Tipul constrângerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_SERVICIU	id_serviciu	Identificatorul unic al serviciului
NOT NULL	—	nume	Fiecare serviciu trebuie să aibă o denumire
	—	specializare	Fiecare serviciu face parte dintr-o specializare
	—	pret	Fiecare serviciu trebuie să aibă un preț
	—	durata	Fiecare serviciu trebuie să aibă o durată
CHECK	SPECIALIZARE_CK	specializare IN ('stom_gen', 'ortodontie', 'implantologie', 'estetica', 'endodontie', 'pedodontie', 'protetica', 'chirurgie', 'radiologie', 'profilaxie')	Specializarea trebuie să fie una dintre cele enumerate, oferite de clinică
	PRET_SERVICIU_CK	pret > 0	Prețul este un număr pozitiv
	DURATA_CK	durata > 0 AND durata <= 60	Durata unui serviciu este un număr pozitiv, mai mic sau egal cu 60
UNIQUE	DENUMIRE_SERV_UQ	denumire	Fiecare serviciu are o denumire unică

7.2. Tabelul PACIENTI

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_PACIENT	id_pacient	Identificatorul unic al pacientului
NOT NULL	—	nume	Fiecare pacient trebuie să aibă un nume de familie
	—	prenume	Fiecare pacient trebuie să aibă un prenume
	—	data_nasterii	Fiecare pacient trebuie să aibă o dată de naștere
	—	telefon	Fiecare pacient trebuie să aibă un număr de telefon
	—	email	Fiecare pacient trebuie să aibă o adresă de email
CHECK	EMAIL_PACIENT_CK	email LIKE '%@%.%'	Emailul pacientului trebuie să respecte structura unui email (trebuie să conțină '@' urmat de '.')

7.3. Tabelul JOBURI

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_JOBURI	id_job	Identificatorul unic al jobului
NOT NULL	—	titlu	Fiecare job trebuie să aibă un titlu
	—	salariu_min	Fiecare job trebuie să aibă un salariu minim
	—	salariu_max	Fiecare job trebuie să aibă un salariu maxim
CHECK	SALARIU_MAX_CK	salariu_max >= salariu_min	Salariul maxim al unui job trebuie să fie mai mare sau egal cu salariul minim pentru acel job
	STUDII_CK	studii_necesare IN ('medii', 'superioare')	Studiile minime necesare pot fi medii sau superioare
UNIQUE	TITLU_JOB_UQ	titlu	Fiecare job are un titlu unic

7.4. Tabelul ANGAJATI

Tipul constrangerii	Denumirea constrangerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_ANGAJAT	id_angajat	Identificatorul unic al angajatului
FOREIGN KEY	FK_JOB_ANGAJAT	(id_job) REFERENCES joburi (id_job)	Coloana id_job referă coloana id_job din tabelul JOBURI
	FK_MANAGER	(id_manager) REFERENCES angajati (id_angajat)	Coloana id_manager referă coloana id_angajat din tabelul ANGAJATI; Managerul unui angajat este tot un angajat
NOT NULL	—	id_job	Fiecare angajat trebuie să aibă un job
	—	nume	Fiecare angajat trebuie să aibă un nume de familie
	—	prenume	Fiecare angajat trebuie să aibă un prenume
	—	data_nasterii	Fiecare angajat trebuie să aibă o dată de naștere
	—	telefon	Fiecare angajat trebuie să aibă un număr de telefon
	—	email	Fiecare angajat trebuie să aibă o adresă de email
CHECK	EMAIL_ANGAJAT_CK	email LIKE '%@%.%'	Emailul angajatului trebuie să respecte structura unui email (trebuie să conțină '@' urmat de '.')
	SALARIU_ANGAJAT_CK	salariu > 0	Salariul este un număr pozitiv
UNIQUE	TELEFON_ANGAJAT_UQ	telefon	Fiecare angajat are un număr de telefon unic
	EMAIL_ANGAJAT_UQ	email	Fiecare angajat are o adresă de email unică

7.5. Tabelul SALI

Tipul constrangerii	Denumirea constrangerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_SALA	id_sala	Identificatorul unic al salii
NOT NULL	—	tip	Fiecare sală este de un anumit tip
	—	suprafață	Fiecare sală are o anumită suprafață
	—	etaj	Fiecare sală este situată pe un anumit etaj

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
CHECK	TIP_CK	tip IN ('cabinet', 'laborator', 'birou', 'recepție')	Tipul sălii trebuie să fie unul dintre următoarele: cabinet, laborator, birou, recepție
	SUPRAFATA_CK	suprafata > 0	Suprafața este un număr pozitiv
	ETAJ_CK	etaj >= 0 AND etaj <= 2	Etajul se încadrează între 0 (parter) și 2

7.6. Tabelul FURNIZORI

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_FURNIZOR	id_furnizor	Identificatorul unic al furnizorului
NOT NULL	—	nume	Fiecare furnizor are un nume
	—	telefon	Fiecare furnizor trebuie să aibă un număr de telefon
	—	email	Fiecare furnizor trebuie să aibă o adresă de email
CHECK	EMAIL_FURN_CK	email LIKE '%@%.%'	Emailul furnizorului trebuie să respecte structura unui email (trebuie să conțină @ urmat de .)
UNIQUE	NUME_FURNIZOR_UQ	nume	Fiecare furnizor are un nume unic
	TELEFON_FURN_UQ	telefon	Fiecare furnizor are un număr de telefon unic
	EMAIL_FURN_UQ	email	Fiecare furnizor are o adresă de email unică

7.7. Tabelul ECHIPAMENTE

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_ECHIPAMENT	id_echipament	Identificatorul unic al echipamentului
NOT NULL	—	denumire	Fiecare echipament trebuie să aibă o denumire
	—	categorie	Fiecare echipament trebuie să facă parte dintr-o categorie
	—	pret	Fiecare echipament trebuie să aibă un preț
CHECK	PRET_ECHIP_CK	pret > 0	Prețul este un număr pozitiv
UNIQUE	DENUMIRE_ECHIP_UQ	denumire	Fiecare echipament are o denumire unică

7.8. Tabelul PROGRAMARI

Tipul constrangerii	Denumirea constrangerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_PROGRAMARI	id_programare	Identificatorul unic al unei programări
FOREIGN KEY	FK_PACIENT_PROG	(id_pacient) REFERENCES pacienti(id_pacient)	Coloana id_pacient referă coloana id_pacient din tabelul PACIENTI
	FK_SERVICIU_PROG	(id_serviciu) REFERENCES servicii(id_serviciu)	Coloana id_serviciu referă coloana id_serviciu din tabelul SERVICII
	FK_ANGAJAT_PROG	(id_angajat) REFERENCES angajati(id_angajat)	Coloana id_angajat referă coloana id_angajat din tabelul ANGAJATI
NOT NULL	—	data_programare	Fiecare programare trebuie să aibă o dată
	—	ora	Fiecare programare trebuie să aibă o oră
CHECK	ORA_PROG_CK	ora >= 9 AND ora <= 16	Clinica are program de la 9-17, iar <i>ora</i> reprezintă ora de început a programării; Programările se fac la oră fixă
UNIQUE	PROGRAMARE_UQ	(id_angajat, data_programare, ora)	Pentru a nu se suprapune programările, tuplul (angajat, data, ora) trebuie să fie unic

7.9. Tabelul REPARTIZARE_SALI

Tipul constrangerii	Denumirea constrangerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_REPARTIZARE	id_repartizare	Identificatorul unic al repartizării
FOREIGN KEY	FK_ANGAJAT_REP	(id_angajat) REFERENCES angajati(id_angajat)	Coloana id_angajat referă coloana id_angajat din tabelul ANGAJATI
	FK_SALA_REP	(id_sala) REFERENCES sali(id_sala)	Coloana id_sala referă coloana id_sala din tabelul SALI
NOT NULL	—	id_sala	Fiecare repartizare trebuie să fie făcută pentru o sală
	—	data_repartizare	Fiecare repartizare trebuie să conțină data
UNIQUE	REPARTIZARE_UQ	(id_sala, data_repartizare)	Repartizările se fac pe zi (sala este ocupată un singur angajat toată ziua)

7.10. Tabelul INVENTARE

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_INVENTAR	id_inventar	Identificatorul unic al inventarului
FOREIGN KEY	FK_SALA_INVENTAR	(id_sala) REFERENCES sali(id_sala)	Coloana id_sala referă coloana id_sala din tabelul SALI
	FK_ECHIP_INVENTAR	(id echipament) REFERENCES echipamente(id echipament)	Coloana id_echipament referă coloana id_echipament din tabelul ECHIPAMENTE
NOT NULL	—	data_inventar	Fiecare inventar trebuie să conțină data
	—	cantitate	Fiecare inventar trebuie să conțină cantitatea
CHECK	CANTITATE_INVENTAR_CK	cantitate > 0	Cantitatea este un număr pozitiv

7.11. Tabelul LIVRARI

Tipul constrangerii	Denumirea constrângerii	Coloanele (sau verificările) la care se referă constrângerile	Explicații
PRIMARY KEY	PK_LIVRARE	id_livrare	Identificatorul unic al livrării
FOREIGN KEY	FK_ECHIP_LIV	(id echipament) REFERENCES echipamente(id echipament)	Coloana id_echipament referă coloana id_echipament din tabelul ECHIPAMENTE
	FK_FURN_LIV	(id_furnizor) REFERENCES furnizori(id_furnizor)	Coloana id_furnizor referă coloana id_furnizor din tabelul FURNIZORI
NOT NULL	—	data_livrare	Fiecare livrare trebuie să se realizeze la o anumită dată
	—	cantitate	Fiecare livrare trebuie să conțină cantitatea
CHECK	CANT_LIVRARE_CK	cantitate > 0	Cantitatea este un număr pozitiv

8. Schemele relaționale

SERVICII (id_serviciu#, denumire, specializare, pret, durata);

PACIENTI (id_pacient#, nume, prenume, data_nasterii, telefon, email);

JOBURI (id_job#, titlu, salariu_min, salariu_max, studii_necesare);

ANGAJATI (id_angajat#, id_job, id_manager, nume, prenume, data_nasterii, telefon, email, salariu);

SALI (id_sala#, tip, suprafata, etaj);

FURNIZORI (id_furnizor#, nume, telefon, email);

ECHIPAMENTE (id_echipament#, denumire, categorie, pret);

PROGRAMARI (id_programare#, id_pacient, id_serviciu, id_angajat, data_programare, ora);

REPARTIZARE_SALI (id_repartizare#, id_angajat, id_sala, data_repartizare);

INVENTARE (id_inventar#, id_sala, id_echipament, data_inventar, cantitate);

LIVRARI (id_livrare#, id_echipament, id_furnizor, data_livrare, cantitate);

9. Constrângerile de ștergere

Constrângerile de ștergere se aplică cheilor externe, conform constrângerilor NOT NULL. Dacă o coloană trebuie să fie diferită de *null*, atunci la ștergerea ei se va folosi o constrângere de tipul *on delete cascade*, prin care se șterge întreaga înregistrare. Altfel, coloanei i se aplică *on delete set null*.

- La ștergerea unui serviciu, coloana id_serviciu din tabelul PROGRAMARI va fi setată la null pentru programările făcute pentru serviciul șters (*on delete set null*).
- La ștergerea unui pacient, coloana id_pacient din tabelul PROGRAMARI va fi setată la null pentru programările făcute pentru pacientul șters (*on delete set null*).
- La ștergerea unui angajat, coloana id_angajat din tabelul PROGRAMARI va fi setată la null pentru programările în care a lucrat angajatul șters (*on delete set null*). Dacă acest angajat este managerul altor angajați, coloana id_manager din tabelul ANGAJATI va fi setată la null pentru angajații respectivi (*on delete set null*). De asemenea, coloana id_angajat din tabelul REPARTIZARE_SALI va fi setată la null pentru repartizările făcute pentru angajatul șters (*on delete set null*).
- La ștergerea unui job, toți angajații care dețin jobul respectiv vor fi șterși (*on delete cascade*). Ștergerea acestor angajați poate duce la ștergerea altor date, conform constrângerilor de mai sus.
- La ștergerea unei săli, toate repartizările făcute pentru sala respectivă vor fi șterse (*on delete cascade*).
- La ștergerea unui echipament, coloana id_echipament din tabelul INVENTARE și coloana id_echipament din tabelul LIVRARI vor fi setate la null pentru inventarele și livrările care conțineau echipamentul șters (*on delete set null*).
- La ștergerea unui furnizor, coloana id_furnizor din tabelul LIVRARI va fi setată la null pentru livrările realizate de furnizorul șters (*on delete set null*).

10. Crearea tabelelor

Ordinea în care se creează tabelele contează, deoarece tabelele care conțin chei externe trebuie să refere coloane din tabele deja existente. Toate constrângerile menționate anterior trebuie să se regăsească în implementarea tabelor. Dacă tabelele există deja, acestea vor fi șterse folosind comanda *drop table if exists*.

```
DROP TABLE IF EXISTS programari;
DROP TABLE IF EXISTS repartizare_sali;
DROP TABLE IF EXISTS inventare;
DROP TABLE IF EXISTS livrari;
DROP TABLE IF EXISTS furnizori;
DROP TABLE IF EXISTS echipamente;
DROP TABLE IF EXISTS sali;
DROP TABLE IF EXISTS angajati;
DROP TABLE IF EXISTS joburi;
DROP TABLE IF EXISTS servicii;
DROP TABLE IF EXISTS pacienti;
```

10.1. Tabelul SERVICII

```
CREATE TABLE servicii
(
  id_serviciu INTEGER,
  denumire VARCHAR(50) NOT NULL,
  specializare VARCHAR(20) NOT NULL,
  pret DOUBLE NOT NULL,
  durata INTEGER NOT NULL,

  CONSTRAINT PK_SERVICIU PRIMARY KEY (id_serviciu),
  CONSTRAINT DENUMIRE_SERV_UQ UNIQUE (denumire),
  CONSTRAINT SPECIALIZARE_CK CHECK (specializare IN
    ('stom_gen', 'ortodontie', 'implantologie', 'estetica', 'endodontie',
    'pedodontie', 'protetica', 'chirurgie', 'radiologie', 'profilaxie')),
  CONSTRAINT PRET_SERVICIU_CK CHECK (pret > 0),
  CONSTRAINT DURATA_CK CHECK (durata > 0 AND durata <= 60)
);
```

10.2. Tabelul PACIENTI

```
CREATE TABLE pacienti
(
  id_pacient INTEGER,
  nume VARCHAR(20) NOT NULL,
  prenume VARCHAR(20) NOT NULL,
  data_nasterii DATE NOT NULL,
  telefon VARCHAR(10) NOT NULL,
  email VARCHAR(40) NOT NULL

  CONSTRAINT PK_PACIENT PRIMARY KEY (id_pacient),
  CONSTRAINT EMAIL_PACIENT_CK CHECK (email LIKE '%@%.%')
);
```

10.3. Tabelul JOBURI

```
CREATE TABLE joburi
(
  id_job INTEGER,
  titlu VARCHAR(30) NOT NULL,
  salariu_min INTEGER NOT NULL,
  salariu_max INTEGER NOT NULL,
  studii_necesare VARCHAR(20),

  CONSTRAINT PK_JOB PRIMARY KEY (id_job),
  CONSTRAINT TITLU_JOB_UQ UNIQUE (titlu),
  CONSTRAINT STUDII_CK CHECK (studii_necesare IN ('medii', 'superioare')),
  CONSTRAINT SALARIU_MAX_CK CHECK (salariu_max >= salariu_min)
);
```

10.4. Tabelul ANGAJATI

```
CREATE TABLE angajati
(
  id_angajat INTEGER,
  id_job INTEGER NOT NULL,
  id_manager INTEGER,
  nume VARCHAR(20) NOT NULL,
  prenume VARCHAR(20) NOT NULL,
  data_nasterii DATE NOT NULL,
  telefon VARCHAR(10) NOT NULL,
  email VARCHAR(40) NOT NULL,
  salariu DOUBLE NOT NULL,

  CONSTRAINT PK_ANGAJAT PRIMARY KEY (id_angajat),
  CONSTRAINT FK_JOB_ANGAJAT FOREIGN KEY (id_job) REFERENCES
    joburi(id_job) ON DELETE CASCADE,
  CONSTRAINT FK_MANAGER FOREIGN KEY (id_manager) REFERENCES
    angajati(id_angajat) ON DELETE SET NULL
  CONSTRAINT TELEFON_ANGAJAT_UQ UNIQUE (telefon),
  CONSTRAINT EMAIL_ANGAJAT_UQ UNIQUE (email),
  CONSTRAINT EMAIL_ANGAJAT_CK CHECK (email LIKE '%@%.%'),
  CONSTRAINT SALARIU_VALID_CK CHECK (salariu > 0)
);
```

10.5.Tabelul SALI

```
CREATE TABLE sali
(
  id_sala INTEGER,
  tip VARCHAR(10) NOT NULL,
  suprafata DOUBLE NOT NULL,
  etaj INTEGER NOT NULL,

  CONSTRAINT PK_SALA PRIMARY KEY (id_sala),
  CONSTRAINT TIP_CK CHECK (tip IN
    ('cabinet', 'laborator', 'birou', 'receptie')),
  CONSTRAINT SUPRAFATA_CK CHECK (suprafata > 0),
  CONSTRAINT ETAJ_CK CHECK (etaj >= 0 AND etaj <= 2)
);
```

10.6.Tabelul FURNIZORI

```
CREATE TABLE furnizori
(
  id_furnizor INTEGER,
  nume VARCHAR(30) NOT NULL,
  telefon VARCHAR(10) NOT NULL,
  email VARCHAR(30) NOT NULL,

  CONSTRAINT PK_FURNIZOR PRIMARY KEY (id_furnizor),
  CONSTRAINT NUME_FURNIZOR_UQ UNIQUE (nume),
  CONSTRAINT TELEFON_FURN_UQ UNIQUE (telefon),
  CONSTRAINT EMAIL_FURN_UQ UNIQUE (email),
  CONSTRAINT EMAIL_FURN_CK CHECK (email LIKE '%@%.%')
);
```

10.7.Tabelul ECHIPAMENTE

```
CREATE TABLE echipamente
(
  id_echipament INTEGER,
  denumire VARCHAR(50) NOT NULL,
  categorie VARCHAR(30) NOT NULL,
  pret DOUBLE NOT NULL,

  CONSTRAINT PK_ECHIPAMENT PRIMARY KEY (id_echipament),
  CONSTRAINT DENUMIRE_ECHIP_UQ UNIQUE (denumire),
  CONSTRAINT PRET_ECHIP_CK CHECK (pret > 0)
);
```

10.8. Tabelul PROGRAMARI

```
CREATE TABLE programari
(
  id_programare INTEGER,
  id_pacient INTEGER,
  id_serviciu INTEGER,
  id_angajat INTEGER,
  data_programare DATE NOT NULL,
  ora INTEGER NOT NULL,

  CONSTRAINT PK_PROGRAMARI PRIMARY KEY (id_programare),
  CONSTRAINT FK_PACIENT_PROG FOREIGN KEY (id_pacient) REFERENCES
    pacienti(id_pacient) ON DELETE SET NULL,
  CONSTRAINT FK_SERVICIU_PROG FOREIGN KEY (id_serviciu) REFERENCES
    servicii(id_serviciu) ON DELETE SET NULL,
  CONSTRAINT FK_ANGAJAT_PROG FOREIGN KEY (id_angajat) REFERENCES
    angajati(id_angajat) ON DELETE SET NULL,
  CONSTRAINT PROGRAMARE_UQ UNIQUE (id_angajat, data_programare, ora),
  CONSTRAINT ORA_PROG_CK CHECK (ora >= 9 AND ora <= 16)
);
```

10.9. Tabelul REPARTIZARE_SALI

```
CREATE TABLE repartizare_sali
(
  id_repartizare INTEGER,
  id_angajat INTEGER,
  id_sala INTEGER NOT NULL,
  data_repartizare DATE NOT NULL,

  CONSTRAINT PK_REPARTIZARE PRIMARY KEY (id_repartizare),
  CONSTRAINT FK_ANGAJAT_REP FOREIGN KEY (id_angajat) REFERENCES
    angajati(id_angajat) ON DELETE SET NULL,
  CONSTRAINT FK_SALA_REP FOREIGN KEY (id_sala) REFERENCES
    sali(id_sala) ON DELETE CASCADE,
  CONSTRAINT REPARTIZARE_UQ UNIQUE (id_sala, data_repartizare)
);
```

10.10.Tabelul INVENTARE

```
CREATE TABLE inventare
(
  id_inventar INTEGER,
  id_sala INTEGER,
  id_echipament INTEGER,
  data_inventar DATE NOT NULL,
  cantitate INTEGER NOT NULL,

  CONSTRAINT PK_INVENTAR PRIMARY KEY (id_inventar),
  CONSTRAINT FK_SALA_INVENTAR FOREIGN KEY (id_sala) REFERENCES
    sali(id_sala) ON DELETE SET NULL,
  CONSTRAINT FK_ECHIP_INVENTAR FOREIGN KEY (id_echipament) REFERENCES
    echipamente(id_echipament) ON DELETE SET NULL,
  CONSTRAINT CANTITATE_INVENTAR_CK CHECK (cantitate > 0)
);
```

10.11.Tabelul LIVRARI

```
CREATE TABLE livrari
(
  id_livrare INTEGER,
  id_echipament INTEGER,
  id_furnizor INTEGER,
  data_livrare DATE NOT NULL,
  cantitate INTEGER NOT NULL,

  CONSTRAINT PK_LIVRARE PRIMARY KEY (id_livrare),
  CONSTRAINT FK_ECHIP_LIV FOREIGN KEY (id_echipament) REFERENCES
    echipamente(id_echipament) ON DELETE SET NULL,
  CONSTRAINT FK_FURN_LIV FOREIGN KEY (id_furnizor) REFERENCES
    furnizori(id_furnizor) ON DELETE SET NULL,
  CONSTRAINT CANT_LIVRARE_CK CHECK (cantitate > 0)
);
```

11.Inserarea datelor în tabele

Și în acest caz, ordinea introducerii datelor are importanță. De exemplu, înregistrările din tabelul ANGAJAȚI sunt dependente de existența unor înregistrări în tabelul JOBURI, caz în care contează ordinea tabelelor la inserare. De asemenea, unele înregistrări din tabelul ANGAJAȚI pot fi condiționate de existența unui manager (adică a altui angajat), deci și ordinea înserării înregistrărilor din același tabel contează. În plus, înregistrările introduse trebuie să respecte constrângerile de integritate impuse.

11.1. Tabelul SERVICII

```
INSERT INTO servicii
VALUES (200, 'obturatie', 'stom_gen', 150, 20);
INSERT INTO servicii
VALUES (201, 'implant dentar', 'implantologie', 3000, 60);
INSERT INTO servicii
VALUES (202, 'albire', 'estetica', 1000, 30);
INSERT INTO servicii
VALUES (203, 'proteza fixa', 'protetica', 5000, 40);
INSERT INTO servicii
VALUES (207, 'consultatie copii', 'pedodontie', 100, 15);
```

11.2. Tabelul PACIENTI

```
INSERT INTO pacienti
VALUES (1003, 'Vladu', 'Alex', '1984/2/21',
      '0755135413', 'alex.vladu@yahoo.com');
INSERT INTO pacienti
VALUES (1005, 'Simion', 'Alexandra', '1978/8/9',
      '0728212663', 'alexandra.simion@gmail.com');
INSERT INTO pacienti
VALUES (1007, 'Dumitrache', 'Radu', '2006/12/20',
      '0772362311', 'rdumitrache@yahoo.com');
INSERT INTO pacienti
VALUES (1011, 'Albu', 'Robert', '1970/11/3',
      '0798743468', 'ralbu@gmail.com');
INSERT INTO pacienti
VALUES (1014, 'Teodorescu', 'Silviu', '2001/5/8',
      '0723499352', 'steodorescu@gmail.com');
```

11.3. Tabelul JOBURI

```
INSERT INTO joburi
VALUES (10, 'CEO', 8000, 10000, 'superioare');
INSERT INTO joburi
VALUES (11, 'manager', 7000, 9000, 'superioare');
INSERT INTO joburi
VALUES (12, 'medic stom generala', 4500, 6000, 'superioare');
INSERT INTO joburi
VALUES (14, 'medic implantologie', 7000, 9000, 'superioare');
INSERT INTO joburi
VALUES (15, 'medic estetica', 5500, 7000, 'superioare');
INSERT INTO joburi
VALUES (16, 'medic pedodontie', 5000, 7000, 'superioare');
```

11.4. Tabelul ANGAJATI

```
INSERT INTO angajati
VALUES (100, 10, null, 'Stoica', 'Alex', '1974/6/18',
       '0734029494', 'astoica@gmail.com', 8300);
INSERT INTO angajati
VALUES (101, 11, 100, 'Munteanu', 'Ionela', '1980/10/14',
       '0768902436', 'munteanu_ionela@gmail.com', 7000);
INSERT INTO angajati
VALUES (102, 11, 100, 'Sava', 'Delia', '1981/6/21',
       '0767930880', 'delia.sava@gmail.com', 8900);
INSERT INTO angajati
VALUES (104, 15, 101, 'Ionescu', 'Iulia', '1976/12/17',
       '0712343818', 'ionescu_iulia@yahoo.com', 6400);
INSERT INTO angajati
VALUES (105, 12, 102, 'Albu', 'Monica', '1978/11/12',
       '0726410305', 'malbu@yahoo.com', 7000);
INSERT INTO angajati
VALUES (108, 16, 103, 'Iacob', 'Larisa', '1976/6/28', '
       0763973708', 'larisa.iacob@gmail.com', 6400);
INSERT INTO angajati
VALUES (118, 14, 101, 'Stan', 'Cosmina', '1983/12/22',
       '0789816799', 'cstan@gmail.com', 7200);
```

11.5. Tabelul SALI

```
INSERT INTO sali
VALUES (1, 'cabinet', 40, 0);
INSERT INTO sali
VALUES (2, 'cabinet', 20, 0);
INSERT INTO sali
VALUES (3, 'cabinet', 40, 0);
INSERT INTO sali
VALUES (4, 'cabinet', 20, 1);
INSERT INTO sali
VALUES (5, 'cabinet', 30, 1);
```

11.6. Tabelul FURNIZORI

```
INSERT INTO furnizori
VALUES (20, 'DoriotDent', '0222310508', 'office@doriotdent.ro');
INSERT INTO furnizori
VALUES (21, 'DentStore', '0216025201', 'office@dentstore.ro');
INSERT INTO furnizori
VALUES (22, 'HeliosDental', '0229222792', 'office@heliosdental.ro');
INSERT INTO furnizori
VALUES (23, 'Medident', '0218608619', 'office@medident.ro');
INSERT INTO furnizori
VALUES (24, 'TerraDent', '0295887001', 'office@terrarent.ro');
```

11.7.Tabelul ECHIPAMENTE

```
INSERT INTO echipamente
VALUES (50, 'autoclava','sterilizare', 50);
INSERT INTO echipamente
VALUES (51, 'lampa albire', 'echipament albire', 10000);
INSERT INTO echipamente
VALUES (52, 'compozit fotopolimerizabil', 'consumabile', 350);
INSERT INTO echipamente
VALUES (53, 'cleste extractie', 'instrumentar chirurgie', 170);
INSERT INTO echipamente
VALUES (55, 'pachet 5 freze', 'instrumentar', 250);
INSERT INTO echipamente
VALUES (57, 'folie diga', 'consumabile', 55);
```

11.8.Tabelul PROGRAMARI

```
INSERT INTO programari
VALUES (1, 1003, 200, 105, '2021-12-23', 13);
INSERT INTO programari
VALUES (2, 1007, 207, 108, '2021-11-11', 16);
INSERT INTO programari
VALUES (3, 1011, 201, 118, '2021-12-15', 9);
INSERT INTO programari
VALUES (4, 1014, 202, 104, '2021-10-29', 11);
INSERT INTO programari
VALUES (5, 1005, 200, 105, '2021-12-23', 10);
```

11.9.Tabelul REPARTIZARE_SALI

```
INSERT INTO repartizare_sali
VALUES (10, 105, 1, '2021-12-23');
INSERT INTO repartizare_sali
VALUES (13, 118, 2, '2021-12-15');
INSERT INTO repartizare_sali
VALUES (14, 104, 3, '2021-10-29');
INSERT INTO repartizare_sali
VALUES (17, 110, 4, '2021-08-09');
INSERT INTO repartizare_sali
VALUES (18, 108, 3, '2021-11-11');
```

11.10. Tabelul INVENTARE

```
INSERT INTO inventare
VALUES (10, 1, 50, '2021-12-23', 1);
INSERT INTO inventare
VALUES (11, 2, 51, '2021-11-30', 1);
INSERT INTO inventare
VALUES (12, 4, 55, '2021-10-01', 3);
INSERT INTO inventare
VALUES (13, 3, 57, '2021-08-05', 20);
INSERT INTO inventare
VALUES (14, 5, 53, '2021-11-15', 2);
```

11.11. Tabelul LIVRARI

```
INSERT INTO livrari
VALUES (10, 53, 20, '2021-08-03', 10);
INSERT INTO livrari
VALUES (11, 53, 21, '2021-10-02', 8);
INSERT INTO livrari
VALUES (12, 50, 21, '2021-05-05', 2);
INSERT INTO livrari
VALUES (13, 52, 23, '2021-10-10', 5);
INSERT INTO livrari
VALUES (14, 57, 23, '2021-12-02', 15);
```

La finalul introducerii datelor, se folosește comanda *commit* pentru a încheia tranzacția și a salva datele.