UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**PROIECT**

**BAZE DE DATE**

PROFESOR COORDONATOR:

VASILE SILVIU – LAURENȚIU

STUDENT:

ALEXANDRU NORINA-ȘTEFANIA

BUCUREȘTI

2024

UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

**GESTIONAREA INFORMAȚIILOR PENTRU**

**UN CENTRU DE MEDITAȚII**

PROFESOR COORDONATOR:

VASILE SILVIU – LAURENȚIU

STUDENT:

ALEXANDRU NORINA-ȘTEFANIA

BUCUREȘTI

2024

Cuprins

[I. Prezentarea modelului 4](#_Toc155371394)

[1. Descriere 4](#_Toc155371395)

[2. Reguli 5](#_Toc155371396)

[II. Diagrama entitate-relație 6](#_Toc155371397)

[1. Descrierea entităților, atributelor și cheilor 6](#_Toc155371398)

[a. ORASE 6](#_Toc155371399)

[b. LOCATII 7](#_Toc155371400)

[c. SALI 7](#_Toc155371401)

[d. JOBURI 7](#_Toc155371402)

[e. ANGAJAȚI 8](#_Toc155371403)

[f. ELEVI 8](#_Toc155371404)

[g. SEDINTE 9](#_Toc155371405)

[2. Descrierea relațiilor și a cardinalităților 9](#_Toc155371406)

[a. ORASE – LOCATII 9](#_Toc155371407)

[b. LOCATII – SALI 10](#_Toc155371408)

[c. SALI – SEDINTE 10](#_Toc155371409)

[d. JOBURI – ANGAJATI 10](#_Toc155371410)

[e. ANGAJATI – SEDINTE 11](#_Toc155371411)

[f. SEDINTE – ELEVI 11](#_Toc155371412)

[III. Diagrama conceptuală 12](#_Toc155371413)

[1. Descrierea constrângerilor de integrabilitate 12](#_Toc155371414)

[a. ORASE 12](#_Toc155371415)

[b. LOCATII 12](#_Toc155371416)

[c. SALI 13](#_Toc155371417)

[d. JOBURI 14](#_Toc155371418)

[e. ANGAJATI 14](#_Toc155371419)

[f. ELEVI 15](#_Toc155371420)

[g. SEDINTE 15](#_Toc155371421)

[h. FACTURI 17](#_Toc155371422)

[2. Scheme relaționale 17](#_Toc155371423)

[IV. Script SQL 18](#_Toc155371424)

[1. Etapa de DROP 18](#_Toc155371425)

[2. Creare tabel ORASE 19](#_Toc155371426)

[3. Creare tabel LOCATII 19](#_Toc155371427)

[4. Creare tabel SALI 19](#_Toc155371428)

[5. Creare tabel JOBURI 20](#_Toc155371429)

[6. Creare tabel ANGAJATI 20](#_Toc155371430)

[7. Creare tabel ELEVI 21](#_Toc155371431)

[8. Creare tabel SEDINTE 22](#_Toc155371432)

[9. Creare tabel FACTURI 23](#_Toc155371433)

[10. Popularea tabelului ORASE 24](#_Toc155371434)

[11. Popularea tabelului LOCATII 24](#_Toc155371435)

[12. Popularea tabelului SALI 25](#_Toc155371436)

[13. Popularea tabelului JOBURI 25](#_Toc155371437)

[14. Popularea tabelului ANGAJATI 25](#_Toc155371438)

[15. Popularea tabelului ELEVI 26](#_Toc155371439)

[16. Popularea tabelului SEDINTE 26](#_Toc155371440)

[17. Popularea tabelului FACTURI 27](#_Toc155371441)

1. Prezentarea modelului
2. Descriere

Un centru de meditații reprezintă un spațiu dedicat elevilor ce se confruntă cu provocări în înțelegerea materiilor școlare, celor care doresc să aprofundeze anumite noțiuni, și, în același timp, celor curioși să exploreze domenii noi. Acest mediu educațional oferă sprijin personalizat, adaptat nevoilor individuale ale elevilor, indiferent de nivelul de cunoștințe.

Elevii beneficiază de sesiuni de meditații care acoperă o gamă variată de materii, având oportunitatea să abordeze subiecte complexe sau să clarifice orice aspecte mai dificile. Profesorii furnizează explicații detaliate și ghidare atentă, facilitând înțelegerea profundă a materiei. Aceste sesiuni nu sunt doar un mijloc de depășire a dificultăților academice, ci și un mod de a încuraja explorarea și descoperirea de cunoștințe noi.

Am ales ca temă a acestui proiect un centru de meditații, datorită experienței mele ca profesor implicat în diverse astfel de activități. Fiindu-mi familiar modul de funcționare al acestor spații educaționale, am fost inspirată să explorez aspectele practice ale gestionării informațiilor din cadrul lor. Motivația mea pentru alegerea acestui subiect provine, în parte, din convingerea că fiecare elev are un ritm și un stil propriu de învățare, iar aceste centre reprezintă o platformă crucială în oferirea sprijinului personalizat de care elevii au nevoie.

1. Reguli

Ca orice model din viața reală, pentru a putea fi reprezentat sub forma unei baze de date, acesta are nevoie de câteva reguli suplimentare.

Înainte de a introduce un nou Angajat, trebuie să existe jobul pe care dorim să i-l atribuim.

Înainte de a crea noi locații, trebuie să existe în baza de date orașul în care acestea se află.

Înainte de a adăuga noi Săli în baza de date, trebuie să existe locația în care acestea se află.

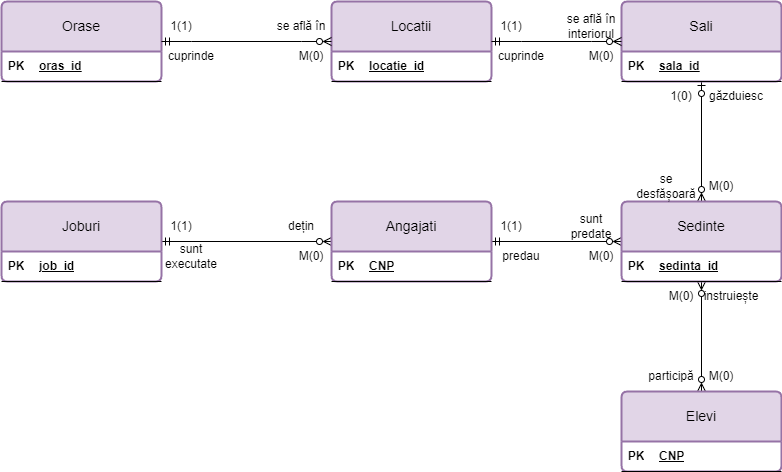
Înainte de a crea noi Ședințe, trebuie să existe angajatul(profesorul) care va desfășura acea lecție și sala (dacă ședința se desfășoară fizic) în care va avea loc ședința. Fiecare ședința va începe la oră fixă și va avea durata de o oră, în intervalul orar 8:00 – 20:00.

Introducerea în baza de date a unei ședințe se face de către profesor(angajat) și acesta alege sala în care se va desfășura activitatea.

Un profesor poate preda o singură materie, deci implicit poate avea un singur job.

Deoarece Angajatii și Elevii sunt identificați prin CNP, nu pot fi înregistrate persoane care nu sunt cetățeni români.

1. Diagrama entitate-relație



1. Descrierea entităților, atributelor și cheilor
2. ORASE

Entitatea Orase conține informații despre orașele în care centrul de meditații are locații. Fiecare oraș este identificat în mod unic prin oras\_id, deoarece pot exista mai multe orașe cu același nume. Astfel, când un nou oraș este introdus în baza de date, i se atribuie un cod unic prin oras\_id.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | oras\_id | NUMBER(5, 0) | Cod unic atribuit fiecărui oraș |
|  | denumire\_oras | VARCHAR2(30 byte) | Numele orașului. |
|  | Judet | VARCHAR2(2 byte) | Codul de maxim 2 litere al județului în care se află orașul. |

1. LOCATII

Entitatea Locatii conține informații despre locațiile în care centrul de meditații își desfășoară activitatea. Prin locații se înțelege clădirea identificată pe hartă prin numele străzii, numărul și blocul, dar și prin indexul orașului în care se află.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | locatie\_id | NUMBER(5, 0) | Cod unic atribuit fiecărei locații |
|  | strada | VARCHAR2(30 byte) | Numele străzii pe care se află locația. |
|  | numar\_locatie | NUMBER(4, 0) | Numărul la care se află locația. |
|  | bloc\_numar | VARCHAR(5 byte) | Opțional, dacă adresa conține și număr și bloc. |
| FK | oras\_id | NUMBER(5, 0) | Indexul orașului |

1. SALI

Entitatea Sali conține informații despre sălile în care se desfășoară ședințele fizice de meditații. Se identifică în mod unic printr-un index generat la adăugarea în baza de date.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | sala\_id | NUMBER(5, 0) | Cod unic atribuit fiecărei săli |
|  | etaj | NUMBER(2, 0) | Etajul din interiorul locației unde se află sala |
|  | numar\_sala | NUMBER(3, 0) | Numărul sălii |
| FK | locatie\_id | NUMBER(5, 0) | Indexul locației |
|  | tip | VARCHAR2(25 byte) | Se referă la dotarea sălii. |

1. JOBURI

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | job\_id | NUMBER(5, 0) | Cod unic atribuit fiecărui job |
|  | denumire\_job | VARCHAR2(15 byte) | Numele jobului |
|  | materie\_predata | VARCHAR2(20 byte) | Denumirea materiei predată pentru fiecare profesor. |
|  | salariu\_minim | NUMBER(8, 2) | Salariu minim primit pentru acest job. |
|  | salariu\_maxim | NUMBER(8, 2) | Salariu maxim primit pentru acest job. |

Entitatea Joburi conține informații despre tipul de joburi pe care le pot avea angajații. Pentru cei care sunt profesori este necesară completarea câmpului materie\_predata, care reprezintă materia pe care o predă fiecare profesor.

1. ANGAJAȚI

Entitatea Angajati memorează informații despre angajații firmei, identificați în mod unic prin CNP.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | CNP | VARCHAR2(13 byte) | CNP-ul persoanei angajate |
|  | nume\_angajat | VARCHAR2(20 byte) | Numele angajatului. |
|  | prenume\_angajat | VARCHAR2(20 byte) | Prenumele angajatului. |
|  | data\_angajarii | DATE | Data la care s-a semnat contractul de angajare. |
|  | salariu | NUMBER(8, 2) | Salariul pe care îl încasează angajatul. |
|  | numar\_telefon | VARCHAR2(10 byte) | Numărul de telefon al angajatului. |
| FK | job\_id | NUMBER | Indexul jobului pe care îl deține. |

1. ELEVI

Entitatea Elevi conține informații despre elevii înscriși la centru. Se consideră că niciunul dintre elevi nu a rămas repetent, iar diferența dintre anul curent și anul începerii ciclului primar (clasa 0) reprezintă clasa în care elevul se află în prezent.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | CNP | VARCHAR2(13 byte) | CNP-ul elevului |
|  | nume\_elev | VARCHAR2(20 byte) | Numele elevului. |
|  | prenume\_elev | VARCHAR2(20 byte) | Prenumele elevului. |
|  | anul\_inscrierii | DATE | Data la care elevul a început clasa 0. |
|  | numar\_telefon\_parinte | VARCHAR2(10 byte) | Numărul de telefon al părintelui elevului. |

1. SEDINTE

Entitatea Sedinte conține informații despre ședințele care se desfășoară în centrul de meditații.

|  |  |  |  |
| --- | --- | --- | --- |
| Cheie | Denumire atribut | Tip de date | Descriere |
| PK | sedinta\_id | NUMBER(5, 0) | Cod unic atribuit fiecărei ședințe. |
|  | sed\_online | VARCHAR2(1 byte) | Dacă ședința este online, se va memora un link, iar dacă este fizic, sala. |
|  | link\_meet | VARCHAR2(40 byte) | Linkul pentru conectarea la ședința online. |
| FK | sala\_id | NUMBER | Indexul sălii dacă ședința se desfășoară fizic |
|  | data\_ora | DATE | Data și ora la care are loc ședința. |
|  | pret | NUMBER(5, 2) | Prețul ședinței pentru un elev. |
| FK | profesor | VARCHAR2(13 byte) | CNP-ul angajatului care desfășoară ședința. |

1. Descrierea relațiilor și a cardinalităților
2. ORASE – LOCATII

**Reguli**

Într-un oraș pot exista mai multe locații.

Într-un oraș este posibil să nu existe nicio locație.

O locație se află maxim într-un oraș.

O locație trebuie să se afle într-un oraș.

**Cardinalități**

ORASE 1(1) – LOCATII M(0)

1. LOCATII – SALI

**Reguli**

Într-o locație pot exista mai multe săli.

Într-o locație este posibil să nu existe nicio sală.

O sală se află în maxim o locație.

O sală trebuie să se afle într-o locație.

**Cardinalități**

LOCATII 1(1) – SALI M(0)

1. SALI – SEDINTE

**Reguli**

Într-o sală se pot desfășura mai multe ședințe.

Într-o sală poate să nu aibă loc nicio ședință.

O ședință se poate ține în maxim o sală.

O ședință poate să nu aibă loc în nicio sală (dacă aceasta este online).

**Cardinalități**

SALI 1(0) – SEDINTE M(0)

1. JOBURI – ANGAJATI

**Reguli**

Un job poate fi deținut de mai mulți angajați.

Un job poate să nu fie deținut de niciun angajat.

Un angajat execută cel mult un job.

Un angajat trebuie să execute cel puțin un job.

**Cardinalități**

JOBURI 1(1) – ANGAJATI M(0)

1. ANGAJATI – SEDINTE

**Reguli**

Un angajat poate să predea mai multe ședințe.

Un angajat poate să nu predea nicio ședință.

O ședință poate fi predată de cel mult un angajat.

O ședință trebuie să fie predată de cel puțin un angajat.

**Cardinalități**

ANGAJATI 1(1) – SEDINTE M(0)

1. SEDINTE – ELEVI

**Reguli**

O ședință poate avea mai mulți elevi.

O ședință poate să nu aibă niciun elev.

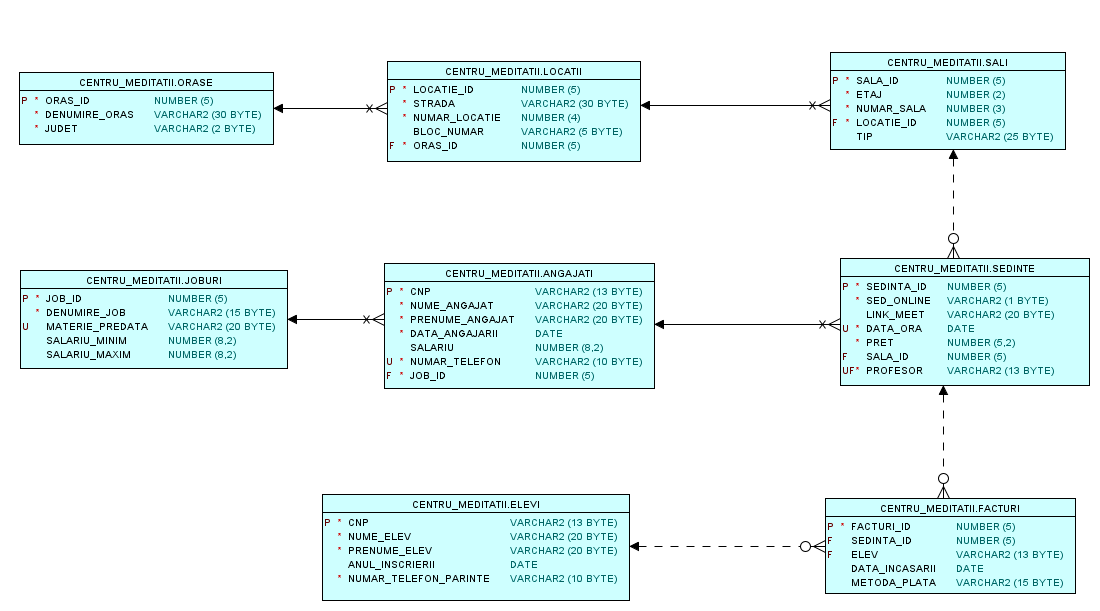
Un elev poate participa la mai multe ședințe.

Un elev poate să nu participe la nicio ședință.

**Cardinalități**

SEDINTE M(0) – ELEVI M(0)

1. Diagrama conceptuală



1. Descrierea constrângerilor de integrabilitate
2. ORASE

**CONSTRÂNGERI DE TIP PRIMARY KEY**

ORASE\_ORAS\_ID\_PK pentru atributul oras\_id

**CONSTRÂNGERI DE TIP NOT NULL**

ORASE\_DENUMIRE\_NN pentru atributul denumire\_oras (denumirea orașului este obligatorie)

ORASE\_JUDET\_NN pentru atributul judet (județul orașului este obligatoriu)

1. LOCATII

**CONSTRÂNGERI DE TIP PRIMARY KEY**

LOCATII\_LOC\_ID\_PK pentru atributul locatie\_id

**CONSTRÂNGERI DE TIP FOREIGN KEY**

LOCATII\_ORAS\_ID\_FK pentru atributul oras\_id, cu referire la atributul oras\_id din tabelul ORASE, creând relația dintre cele 2 tabele

ON DELETE CASCADE: când un oraș este șters, sunt șterse și toate locațiile care se aflau în acel oraș; se păstrează cardinalitatea minimală a relației – ORASE (1) – LOCATII (0)

**CONSTRÂNGERI DE TIP NOT NULL**

LOCATII\_ORAS\_ID\_NN pentru atributul oras\_id (orașul de care aparține locația este obligatorie)

LOCATII\_STRADA\_NN pentru atributul strada (strada pe care se află locația este obligatorie)

LOCATII\_NUMAR\_NN pentru atributul numar\_locatie (numărul pe care se află locația este obligatorie)

1. SALI

**CONSTRÂNGERI DE TIP PRIMARY KEY**

SALI\_SALA\_ID\_PK pentru atributul sala\_id

**CONSTRÂNGERI DE TIP FOREIGN KEY**

SALI\_LOCATIE\_ID\_FK pentru atributul locatie\_id, cu referire la atributul locatie\_id din tabelul LOCATII, creând relația dintre cele 2 tabele

ON DELETE CASCADE: când una dintre locații este ștearsă, vor fi șterse toate sălile din acea locație; se păstrează cardinalitatea minimală a relației – LOCATII (1) – SALI (0)

**CONSTRÂNGERI DE TIP NOT NULL**

SALI\_LOCATIE\_ID\_NN pentru atribului locatie\_id (locația în care se află sala este obligatorie)

SALI\_ETAJ\_NN pentru atributul etaj (etajul la care se află sala este obligatoriu)

SALI\_NUMAR\_NN pentru atributul numar\_sala (fiecare sală este identificată pe fiecare etaj print-un număr, care este obligatoriu)

**CONSTRÂNGERI DE TIP CHECK**

SALI\_TIP\_CK pentru atributul tip (verifică dacă tipul sălii este unul dintre următoarele 'laborator informatica', 'laborator stiinte', 'sala de clasa')

1. JOBURI

**CONSTRÂNGERI DE TIP PRIMARY KEY**

JOBURI\_JOB\_ID\_PK pentru atributul job\_id

**CONSTRÂNGERI DE TIP NOT NULL**

JOBURI\_DENUMIRE\_NN pentru atributul denumire\_job (denumirea jobului este obligatoriu)

**CONSTRÂNGERI DE TIP CHECK**

JOBURI\_MATERIE\_CK pentru atributul materie\_predata (poate fi null, pentru persoanele care nu au jobul de profesor, sau una dintre următoarele 'informatica', 'matematica', 'geografie', 'istorie', 'limba romana', 'limba engleza', 'limba franceza', 'fizica', 'chimie', 'biologie')

**CONSTRÂNGERI DE TIP UNIQUE**

JOBURI\_DEN\_MAT\_UQ pentru atributele denumire\_job, materie\_predata (trebuie să existe o singură dată în baza de date perechea denumire\_job, materie\_predata)

1. ANGAJATI

**CONSTRÂNGERI DE TIP PRIMARY KEY**

ANGAJATI\_CNP\_PK pentru atributul CNP

**CONSTRÂNGERI DE TIP FOREIGN KEY**

ANGAJATI\_JOB\_ID\_FK pentru atributul job\_id, cu referire la atributul job\_id din tabelul JOBURI, creând relația dintre cele 2 tabele

ON DELETE CASCADE: când un job este șters, vor fi șterși și toți angajații care ocupau acel job; se păstrează cardinalitatea minimală a relației – JOBURI (1) – ANGAJATI (0)

**CONSTRÂNGERI DE TIP NOT NULL**

ANGAJATI\_JOB\_ID\_NN pentru atributul job\_id (jobul angajatului este obligatoriu)

ANGAJATI\_NUME\_NN pentru atributul nume\_angajat (numele angajatului este obligatoriu)

ANGAJATI\_PRENUME\_NN pentru atributul prenume\_angajat (prenumele angajatului este obligatoriu)

ANGAJATI\_DATE\_NN pentru atributul data\_angajarii (data angajării este obligatorie)

ANGAJATI\_TELEFON\_NN pentru atributul numar\_telefon (numărul de telefon al angajatului este obligatoriu)

**CONSTRÂNGERI DE TIP CHECK**

ANGAJATI\_CNP\_CK pentru atributul CNP (verifică dacă CNP-ul este unul valid: are lungimea de 13 cifre și verifică cifra de control)

**CONSTRÂNGERI DE TIP UNIQUE**

ANGAJATI\_TELEFON\_UQ pentru atributul numar\_telefon (nu pot exista mai mulți angajați cu același număr de telefon)

1. ELEVI

**CONSTRÂNGERI DE TIP PRIMARY KEY**

ELEVI\_CNP\_PK pentru atributul CNP

**CONSTRÂNGERI DE TIP NOT NULL**

ELEVI\_NUME\_NN pentru atributul nume\_elev (numele elevului este obligatoriu)

ELEVI\_PRENUME\_NN pentru atributul prenume\_elev (prenumele elevului este obligatoriu)

ELEVI\_TELEFON\_NN pentru atributul numar\_telefon\_parinte (numărul de telefon al unuia dintre părinții elevului este obligatoriu)

**CONSTRÂNGERI DE TIP CHECK**

ELEVI\_CNP\_CK pentru atributul CNP (verifică dacă CNP este valid, să aibă exact 13 cifre și să se verifice cifra de control)

1. SEDINTE

**CONSTRÂNGERI DE TIP PRIMARY KEY**

SEDINTE\_SED\_ID\_PK pentru atributul sedinta\_id

**CONSTRÂNGERI DE TIP FOREIGN KEY**

SEDINTE\_PROFESOR\_FK pentru atributul profesor, cu referire la atributul CNP din tabelul ANGAJATI, creând relația dintre cele 2 tabele

ON DELETE CASCADE: când un angajat este șters, se vor șterge toate ședințele pe care acesta le desfășoară; se păstrează cardinalitatea minimală a relației – ANGAJATI (1) – SEDINTE (0)

SEDINTE\_SALA\_ID\_FK pentru atributul sala\_id, cu referire la atributul sala\_id din tabelul SALI, creând relația dintre cele 2 tabele

ON DELETE SET NULL: când o sală este ștearsă, nu vor fi eliminate ședințele din baza de date

**CONSTRÂNGERI DE TIP NOT NULL**

SEDINTE\_ONLINE\_NN pentru atributul sed\_online (este obligatorie precizare tipului de ședința)

SEDINTE\_ORA\_NN pentru atributul data\_ora (este obligatorie precizarea datei și a orei la care se desfășoară ședința)

SEDINTE\_PRET\_NN pentru atributul pret (prețul unei ședințe este obligatoriu)

SEDINTE\_PROFESOR\_NN pentru atributul profesor (fiecare ședință trebuie să aibă un profesor atribuit)

**CONSTRÂNGERI DE TIP CHECK**

SEDINTE\_ONLINE\_CK pentru atributul sed\_online (verifică dacă atributul are una dintre valorile 'Y', 'N')

SEDINTE\_ORA\_CK pentru atributul data\_ora (verifică dacă este o ora validă in conformitate cu regulile, ora să fie între 8 și 20 și să aibă minutele și secundele 0 și 0, pentru a începe la fix)

**CONSTRÂNGERI DE TIP UNIQUE**

SEDINTE\_PROFESOR\_ORA\_UQ pentru atributele profesor, data\_ora (un profesor nu poate preda mai mult de o ședința la aceeași ora, perechea trebuie să fie unică)

**INDEXURI DE UNICITATE**

SEDINTE\_LINK\_ORA\_UQ pentru atributele link\_meet, data\_ora (pentru ședințele online, care au link\_meet diferit de null, nu pot exista 2 ședințe la aceeași ora pe același link de meet, deci perechea trebuie să fie unică)

SEDINTE\_SALA\_ORA\_UQ pentru atributele sala\_id, data\_ora (pentru ședințele fizice, care au sala\_id diferit de null, nu pot exista 2 ședințe la aceeași oră în aceeași sala, deci perechea trebuie să fie unică)

1. FACTURI

**CONTRÂNGERI DE TIP PRIMARY KEY**

FACTURI\_FAC\_ID\_PK pentru atributul factura\_id

**CONTRÂNGERI DE TIP FOREIGN KEY**

FACTURI\_ELEV\_FK pentru atributul elev cu referire la atributul CNP din tabelul ELEVI

ON DELETE SET NULL: când un elev este șters, facturile vor fi păstrate în baza de date

FACTURI\_SED\_ID\_FK pentru atributul sedinta\_id cu referire la atributul sedinta\_id din tabelul SEDINTE

ON DELETE SET NULL: când o ședința este ștearsă, facturile vor fi păstrate în baza de date

**CONSTRÂNGERI DE TIP CHECK**

FACTURI\_METODA\_CK pentru atributul metoda\_plata (verifică dacă este una dintre următoarele 'cash', 'card')

**INDEXURI DE UNICITATE**

FACTURI\_SED\_ELEV\_UQ pentru atributele sedinta\_id, elev (dacă ambele atribute sunt diferite de null perechea trebuie să fie unică, adică un elev nu poate plăti de mai multe ori aceeași ședință)

1. Scheme relaționale

ORASE (oras\_id#, denumire\_oras, judet)

LOCATII (locatie\_id#, strada, numar\_locatie, bloc\_numar, oras\_id(FK))

SALI (sala\_id#, etaj, numar\_sala, locatie\_id(FK), tip)

JOBURI (job\_id#, denumire\_job, materie\_predata, salariu\_minim, salariu\_maxim)

ANGAJATI (CNP#, nume\_angajat, prenume\_angajat, data\_angajarii, salariu, numar\_telefon, job\_id(FK))

ELEVI (CNP#, nume\_elev, prenume\_elev, anul\_inscrierii, numar\_telefon\_parinte)

SEDINTE (sedinta\_id#, sed\_online, link\_meet, data\_ora, pret, sala\_id(FK), profesor(FK))

FACTURI (facturi\_id#, sedinta\_id(FK), elev(FK), data\_incasarii, metoda\_plata)

1. Script SQL

Scriptul SQL, salvat cu numele SCRIPT\_ALEXANDRU\_NORINA\_STEFANIA\_GR263.sql, a fost scris în Oracle SQL Developer v23.1, iar pentru baza de date locală am folosit Oracle Database 21c. La prima rulare a scriptului se vor afișa erori datorită operațiilor de DROP de la începutul acestuia, dar crearea tabelelor și popularea acestora se vor face fără probleme.

1. Etapa de DROP

--DROP  
   
--STERGEREA INDEXURILOR SUPLIMENTALE  
DROP INDEX FACTURI\_sed\_elev\_UQ;  
DROP INDEX SEDINTE\_link\_ora\_UQ;  
DROP INDEX SEDINTE\_sala\_ora\_UQ;  
   
--STERGEREA TABELULUI FACTURI  
DROP TABLE FACTURI;  
   
--STERGEREA TABELULUI ELEVI  
DROP TABLE ELEVI;  
   
--STERGEREA TABELULUI SEDINTE  
DROP TABLE SEDINTE;  
   
--STERGEREA TABELULUI ANGAJATI  
DROP TABLE ANGAJATI;  
   
--STERGEREA TABELULUI JOBURI  
DROP TABLE JOBURI;  
   
--STERGEREA TABELULUI SALI  
DROP TABLE SALI;  
   
--STERGEREA TABELULUI LOCATII  
DROP TABLE LOCATII;  
   
--STERGEREA TABELULUI ORASE  
DROP TABLE ORASE;

1. Creare tabel ORASE

-- CREARE TABEL ORASE  
CREATE TABLE ORASE (  
 oras\_id NUMBER(5, 0)   
 CONSTRAINT ORASE\_ORAS\_ID\_PK PRIMARY KEY,  
 denumire\_oras VARCHAR2(30 byte)   
 CONSTRAINT ORASE\_denumire\_NN NOT NULL,  
 judet VARCHAR2(2 byte)  
 CONSTRAINT ORASE\_judet\_NN NOT NULL  
);

1. Creare tabel LOCATII

-- CREARE TABEL LOCATII  
CREATE TABLE LOCATII (  
 locatie\_id NUMBER(5, 0)  
 CONSTRAINT LOCATII\_LOC\_ID\_PK PRIMARY KEY,  
 strada VARCHAR2(30 byte)  
 CONSTRAINT LOCATII\_strada\_NN NOT NULL,  
 numar\_locatie NUMBER(4, 0)  
 CONSTRAINT LOCATII\_numar\_NN NOT NULL,  
 bloc\_numar VARCHAR2(5 byte),  
 oras\_id NUMBER(5, 0)  
 CONSTRAINT LOCATII\_oras\_id\_FK REFERENCES ORASE(oras\_id)   
 ON DELETE CASCADE  
 CONSTRAINT LOCATII\_oras\_id\_NN NOT NULL  
);

1. Creare tabel SALI

-- CREARE TABEL SALI  
CREATE TABLE SALI (  
 sala\_id NUMBER(5, 0)   
 CONSTRAINT SALI\_SALA\_ID\_PK PRIMARY KEY,  
 etaj NUMBER(2, 0)  
 CONSTRAINT SALI\_etaj\_NN NOT NULL,  
 numar\_sala NUMBER(3, 0)  
 CONSTRAINT SALI\_numar\_NN NOT NULL,  
 locatie\_id NUMBER(5, 0)  
 CONSTRAINT SALI\_locatie\_id\_FK REFERENCES LOCATII(locatie\_id)  
 ON DELETE CASCADE  
 CONSTRAINT SALI\_locatie\_id\_NN NOT NULL,  
 tip VARCHAR2(25 byte)   
 DEFAULT 'sala de clasa'  
 CONSTRAINT SALI\_tip\_CK CHECK (tip in ('laborator informatica', 'laborator stiinte', 'sala de clasa'))  
);

1. Creare tabel JOBURI

-- CREARE TABEL JOBURI  
CREATE TABLE JOBURI (  
 job\_id NUMBER(5, 0)  
 CONSTRAINT JOBURI\_JOB\_ID\_PK PRIMARY KEY,  
 denumire\_job VARCHAR2(15 BYTE)  
 CONSTRAINT JOBURI\_denumire\_NN NOT NULL,  
 materie\_predata VARCHAR2(20 byte)  
 CONSTRAINT JOBURI\_materie\_CK CHECK (materie\_predata is null   
 or materie\_predata in ('informatica', 'matematica', 'geografie', 'istorie', 'limba romana', 'limba engleza', 'limba franceza', 'fizica', 'chimie', 'biologie')),  
 salariu\_minim NUMBER(8, 2),  
 salariu\_maxim NUMBER(8, 2),  
 CONSTRAINT JOBURI\_den\_mat\_UQ UNIQUE(denumire\_job, materie\_predata)  
);

1. Creare tabel ANGAJATI

-- CREARE TABEL ANGAJATI  
CREATE TABLE ANGAJATI (  
 CNP VARCHAR2(13 byte)  
 CONSTRAINT ANGAJATI\_CNP\_PK PRIMARY KEY  
 CONSTRAINT ANGAJATI\_CNP\_CK CHECK (  
 length(CNP) = 13   
 and translate(CNP, 'x0123456789', 'x') is null  
 --and to\_date(substr(CNP, 2, 6), 'yymmdd') is not null  
 and to\_number(substr(CNP, 13, 1)) =   
 case  
 when mod(to\_number(substr(CNP, 1, 1)) \* 2 +  
 to\_number(substr(CNP, 2, 1)) \* 7 +  
 to\_number(substr(CNP, 3, 1)) \* 9 +  
 to\_number(substr(CNP, 4, 1)) \* 1 +  
 to\_number(substr(CNP, 5, 1)) \* 4 +  
 to\_number(substr(CNP, 6, 1)) \* 6 +  
 to\_number(substr(CNP, 7, 1)) \* 3 +  
 to\_number(substr(CNP, 8, 1)) \* 5 +  
 to\_number(substr(CNP, 9, 1)) \* 8 +  
 to\_number(substr(CNP, 10, 1)) \* 2 +  
 to\_number(substr(CNP, 11, 1)) \* 7 +  
 to\_number(substr(CNP, 12, 1)) \* 9,   
 11  
 ) = 10 then 1  
 else mod(to\_number(substr(CNP, 1, 1)) \* 2 +  
 to\_number(substr(CNP, 2, 1)) \* 7 +  
 to\_number(substr(CNP, 3, 1)) \* 9 +  
 to\_number(substr(CNP, 4, 1)) \* 1 +  
 to\_number(substr(CNP, 5, 1)) \* 4 +  
 to\_number(substr(CNP, 6, 1)) \* 6 +  
 to\_number(substr(CNP, 7, 1)) \* 3 +  
 to\_number(substr(CNP, 8, 1)) \* 5 +  
 to\_number(substr(CNP, 9, 1)) \* 8 +  
 to\_number(substr(CNP, 10, 1)) \* 2 +  
 to\_number(substr(CNP, 11, 1)) \* 7 +  
 to\_number(substr(CNP, 12, 1)) \* 9,   
 11  
 )  
 end  
 ),  
 nume\_angajat VARCHAR2(20 BYTE)  
 CONSTRAINT ANGAJATI\_nume\_NN NOT NULL,  
 prenume\_angajat VARCHAR2(20 BYTE)  
 CONSTRAINT ANGAJATI\_prenume\_NN NOT NULL,  
 data\_angajarii DATE  
 CONSTRAINT ANGAJATI\_date\_NN NOT NULL,  
 salariu NUMBER(8, 2),  
 numar\_telefon VARCHAR2(10 BYTE)  
 CONSTRAINT ANGAJATI\_telefon\_NN NOT NULL  
 CONSTRAINT ANGAJATI\_telefon\_UQ UNIQUE,  
 job\_id NUMBER(5, 0)  
 CONSTRAINT ANGAJATI\_job\_id\_FK REFERENCES JOBURI(job\_id)  
 ON DELETE CASCADE  
 CONSTRAINT ANGAJATI\_job\_id\_NN NOT NULL   
);

1. Creare tabel ELEVI

--CREAREA TABEL ELEVI  
CREATE TABLE ELEVI (  
 CNP VARCHAR2(13 byte)  
 CONSTRAINT ELEVI\_CNP\_PK PRIMARY KEY  
 CONSTRAINT ELEVI\_CNP\_CK CHECK (  
 length(CNP) = 13   
 and translate(CNP, 'x0123456789', 'x') is null  
 --and to\_date(substr(CNP, 2, 6), 'yymmdd') is not null  
 and to\_number(substr(CNP, 13, 1)) =   
 case  
 when mod(to\_number(substr(CNP, 1, 1)) \* 2 +  
 to\_number(substr(CNP, 2, 1)) \* 7 +  
 to\_number(substr(CNP, 3, 1)) \* 9 +  
 to\_number(substr(CNP, 4, 1)) \* 1 +  
 to\_number(substr(CNP, 5, 1)) \* 4 +  
 to\_number(substr(CNP, 6, 1)) \* 6 +  
 to\_number(substr(CNP, 7, 1)) \* 3 +  
 to\_number(substr(CNP, 8, 1)) \* 5 +  
 to\_number(substr(CNP, 9, 1)) \* 8 +  
 to\_number(substr(CNP, 10, 1)) \* 2 +  
 to\_number(substr(CNP, 11, 1)) \* 7 +  
 to\_number(substr(CNP, 12, 1)) \* 9,   
 11  
 ) = 10 then 1  
 else mod(to\_number(substr(CNP, 1, 1)) \* 2 +  
 to\_number(substr(CNP, 2, 1)) \* 7 +  
 to\_number(substr(CNP, 3, 1)) \* 9 +  
 to\_number(substr(CNP, 4, 1)) \* 1 +  
 to\_number(substr(CNP, 5, 1)) \* 4 +  
 to\_number(substr(CNP, 6, 1)) \* 6 +  
 to\_number(substr(CNP, 7, 1)) \* 3 +  
 to\_number(substr(CNP, 8, 1)) \* 5 +  
 to\_number(substr(CNP, 9, 1)) \* 8 +  
 to\_number(substr(CNP, 10, 1)) \* 2 +  
 to\_number(substr(CNP, 11, 1)) \* 7 +  
 to\_number(substr(CNP, 12, 1)) \* 9,   
 11  
 )  
 end  
 ),  
 nume\_elev VARCHAR2(20 BYTE)  
 CONSTRAINT ELEVI\_nume\_NN NOT NULL,  
 prenume\_elev VARCHAR2(20 BYTE)  
 CONSTRAINT ELEVI\_prenume\_NN NOT NULL,  
 anul\_inscrierii DATE,  
 numar\_telefon\_parinte VARCHAR2(10 BYTE)  
 CONSTRAINT ELEVI\_telefon\_NN NOT NULL  
);

1. Creare tabel SEDINTE

--CREARE TABEL SEDINTE  
CREATE TABLE SEDINTE (  
 sedinta\_id NUMBER(5, 0)  
 CONSTRAINT SEDINTE\_SED\_ID\_PK PRIMARY KEY,  
 sed\_online VARCHAR2(1 BYTE)  
 CONSTRAINT SEDINTE\_online\_CK CHECK (sed\_online in ('Y', 'N'))  
 CONSTRAINT SEDINTE\_online\_NN NOT NULL,  
 link\_meet VARCHAR2(20 BYTE),  
 data\_ora DATE  
 CONSTRAINT SEDINTE\_ora\_NN NOT NULL  
 CONSTRAINT SEDINTE\_ora\_CK CHECK (  
 to\_number(to\_char(data\_ora, 'hh24')) between 8 and 20   
 and to\_char(data\_ora, 'mi:ss') = '00:00'  
 ),  
 pret NUMBER(5, 2)  
 CONSTRAINT SEDINTE\_pret\_NN NOT NULL,  
 sala\_id NUMBER(5, 0)  
 CONSTRAINT SEDINTE\_sala\_id\_FK REFERENCES SALI(sala\_id)  
 ON DELETE SET NULL,  
 profesor VARCHAR2(13 BYTE)  
 CONSTRAINT SEDINTE\_profesor\_FK REFERENCES ANGAJATI(CNP)  
 ON DELETE CASCADE  
 CONSTRAINT SEDINTE\_profesor\_NN NOT NULL,  
   
 CONSTRAINT SEDINTE\_profesor\_ora\_UQ UNIQUE (profesor, data\_ora)  
);  
  
--ADAUGAREA CONSTRANGERII DE UNIQUE PE COLOANELE (data\_ora, sala\_id)   
--DOAR DACA SALA\_ID NU E NULL  
CREATE UNIQUE INDEX SEDINTE\_sala\_ora\_UQ ON SEDINTE (  
 case  
 when sala\_id is not null  
 then sala\_id  
 else null  
 end,  
 case   
 when sala\_id is not null  
 then data\_ora  
 else null  
 end  
);  
  
--ADAUGAREA CONSTRANGERII DE UNIQUE PE COLOANELE (data\_ora,   
--link\_meet) DOAR DACA LINK\_MEET NU E NULL  
CREATE UNIQUE INDEX SEDINTE\_link\_ora\_UQ ON SEDINTE (  
 case   
 when link\_meet is not null  
 then link\_meet  
 else null  
 end,  
 case   
 when link\_meet is not null  
 then data\_ora  
 else null  
 end  
);

1. Creare tabel FACTURI

--CREARE TABEL FACTURI  
CREATE TABLE FACTURI (  
 facturi\_id NUMBER(5, 0)  
 CONSTRAINT FACTURI\_FAC\_ID\_PK PRIMARY KEY,  
 sedinta\_id NUMBER(5, 0)  
 CONSTRAINT FACTURI\_sed\_id\_FK REFERENCES SEDINTE(sedinta\_id)  
 ON DELETE SET NULL,  
 elev VARCHAR2(13 BYTE)  
 CONSTRAINT FACTURI\_elev\_FK REFERENCES ELEVI(CNP)  
 ON DELETE SET NULL,  
 data\_incasarii DATE,  
 metoda\_plata VARCHAR2(15 BYTE)  
 CONSTRAINT FACTURI\_metoda\_CK CHECK (metoda\_plata in ('cash', 'card'))  
);  
  
  
--ADAUGAREA CONSTRANGERII DE UNIQUE PE COLOANELE (sedinta\_id, elev)   
--DOAR DACA AMBELE VALORI NU SUNT NULE  
CREATE UNIQUE INDEX FACTURI\_sed\_elev\_UQ ON FACTURI (  
 case   
 when sedinta\_id is null or elev is null  
 then null  
 else sedinta\_id  
 end,  
 case  
 when sedinta\_id is null or elev is null  
 then null  
 else elev  
 end  
);

1. Popularea tabelului ORASE

--TABELUL ORASE  
insert into ORASE values (10, 'BUCURESTI', 'B');  
insert into ORASE values (20, 'CLUJ-NAPOCA', 'CJ');  
insert into ORASE values (30, 'TIMISOARA', 'TM');  
insert into ORASE values (40, 'IASI', 'IS');  
insert into ORASE values (50, 'GALATI', 'GL');  
insert into ORASE values (60, 'CONSTANTA', 'CT');  
insert into ORASE values (70, 'VASLUI', 'VS');  
insert into ORASE values (80, 'TULCEA', 'TL');  
insert into ORASE values (90, 'ARAD', 'AR');  
insert into ORASE values (100, 'BISTRITA', 'BN');  
insert into ORASE values (110, 'SUCEAVA', 'SV');  
insert into ORASE values (120, 'BARLAD', 'VS');  
insert into ORASE values (130, 'HUSI', 'VS');  
insert into ORASE values (140, 'TECUCI', 'GL');

1. Popularea tabelului LOCATII

--TABELUL LOCATII  
insert into LOCATII values (12, 'Viorele', 52, null, 10);   
insert into LOCATII values (22, 'Pacii', 1, null, 70);  
insert into LOCATII values (32, 'Splaiul Independentei', 204, null, 10);  
insert into LOCATII values (42, 'Turturelelor', 48, null, 10);  
insert into LOCATII values (52, 'Teodor Mihali', 58, null, 20);  
insert into LOCATII values (62, 'Universitatii', 16, null, 40);  
insert into LOCATII values (72, 'Vasile Parvan', 4, null, 30);  
insert into LOCATII values (82, 'Revolutiei', 94, null, 90);  
insert into LOCATII values (92, 'Drum. Tarpiului', 21, null, 100);  
insert into LOCATII values (102, 'Mamaia', 124, null, 60);  
insert into LOCATII values (112, 'Domneasca', 47, null, 50);  
insert into LOCATII values (122, 'Pacii', 10, null, 110);  
insert into LOCATII values (132, 'Victoriei', 97, null, 80);  
insert into LOCATII values (142, 'Decebal', 13, null, 40);  
insert into LOCATII values (152, 'Gradinilor', 1, null, 20);

1. Popularea tabelului SALI

--TABELUL SALI  
insert into SALI values (15, 0, 2, 12, 'sala de clasa');  
insert into SALI values (25, 0, 1, 12, 'sala de clasa');  
insert into SALI values (35, 0, 3, 12, 'sala de clasa');  
insert into SALI values (45, 0, 4, 12, 'sala de clasa');  
insert into SALI values (55, 2, 23, 42, 'laborator informatica');  
insert into SALI values (65, 4, 42, 42, 'sala de clasa');  
insert into SALI values (75, 1, 12, 92, 'sala de clasa');  
insert into SALI values (85, 2, 2, 102, 'laborator stiinte');  
insert into SALI values (95, 1, 4, 22, 'sala de clasa');  
insert into SALI values (105, 2, 7, 22, 'laborator informatica');  
insert into SALI values (115, 0, 17, 52, 'sala de clasa');  
insert into SALI values (125, 3, 122, 142, 'laborator stiinte');  
insert into SALI values (135, 4, 402, 72, 'laborator informatica');  
insert into SALI values (145, 5, 52, 132, 'sala de clasa');  
insert into SALI values (155, -1, 2, 32, 'sala de clasa');  
insert into SALI values (165, 8, 324, 82, 'laborator stiinte');

1. Popularea tabelului JOBURI

--TABEL JOBURI  
insert into JOBURI values (17, 'contabil', null, 2000, 10000);  
insert into JOBURI values (27, 'profesor', 'informatica', 2000, 9000);  
insert into JOBURI values (37, 'profesor', 'matematica', 2000, 9000);  
insert into JOBURI values (47, 'profesor', 'geografie', 2000, 8000);  
insert into JOBURI values (57, 'profesor', 'istorie', 2000, 8000);  
insert into JOBURI values (67, 'profesor', 'limba romana', 2000, 9000);  
insert into JOBURI values (77, 'profesor', 'limba franceza', 2000, 7000);  
insert into JOBURI values (87, 'profesor', 'limba engleza', 2000, 7000);  
insert into JOBURI values (97, 'profesor', 'fizica', 2000, 9000);  
insert into JOBURI values (107, 'profesor', 'biologie', 2000, 8000);  
insert into JOBURI values (117, 'profesor', 'chimie', 2000, 9000);  
insert into JOBURI values (127, 'director', null, 5000, 15000);

1. Popularea tabelului ANGAJATI

--TABEL ANGAJATI  
insert into ANGAJATI values ('1841122411733', 'Popescu', 'Vasile', '12-APR-2021', 6000, '0745232323', 17 );  
insert into ANGAJATI values ('1830901416261', 'Bujor', 'Gheorghe', '10-APR-2021', 10000, '0740434178', 127);  
insert into ANGAJATI values ('1910919412026', 'Munteanu', 'Vasile', '1-MAY-2022', 4500, '0720754743', 57);  
insert into ANGAJATI values ('1751024416058', 'Sandu', 'Catalin', '3-JUN-2021', 4500, '0745035882', 47);  
insert into ANGAJATI values ('1810711417513', 'Balan', 'Sorin', '21-SEP-2021', 5500, '0743044863', 97);  
insert into ANGAJATI values ('1940131415026', 'Moldoveanu', 'Pavel', '15-OCT-2021', 6000, '0735401616', 27);  
insert into ANGAJATI values ('1960714417703', 'Mita', 'Alexandru', '4-NOV-2022', 5000, '0766530642', 77);  
insert into ANGAJATI values ('5010106415045', 'Spataru', 'Dan', '5-SEP-2023', 6500, '0742734510', 37);  
insert into ANGAJATI values ('2890613280853', 'Sava', 'Lucia', '15-MAY-2021', 5500, '0771420110', 67);  
insert into ANGAJATI values ('2921024331691', 'Vulpe', 'Irina', '6-JUN-2022', 6500, '0743847669', 27);  
insert into ANGAJATI values ('2870906230977', 'Sima', 'Antonia', '12-DEC-2021', 5000, '0741669324', 87);  
insert into ANGAJATI values ('2760924018506', 'Radu', 'Mihaela', '21-JAN-2022', 5250, '0764355422', 107);  
insert into ANGAJATI values ('2960805527011', 'Ciobanu', 'Alina', '10-FEB-2023', 6500, '0743950351', 37);  
insert into ANGAJATI values ('2771031152107', 'Mitache', 'Dorina', '12-SEP-2022', 4000, '0741455809', 67);  
insert into ANGAJATI values ('2840930240988', 'Selaru', 'Valerica', '10-OCT-2023', 5000, '0799770448', 117);  
insert into ANGAJATI values ('6000620090130', 'Necula', 'Ana', '11-NOV-2023', null, '0745886573', 107);

1. Popularea tabelului ELEVI

--TABEL ELELVI  
insert into ELEVI values ('5100217419827', 'Popa', 'Andrei', '01-SEP-2016', '0740015893');  
insert into ELEVI values ('5110523111179', 'Popa', 'Mihai', '01-SEP-2017', '0740015893');  
insert into ELEVI values ('5150705035250', 'Onica', 'Victor', '01-SEP-2021', '0746019265');  
insert into ELEVI values ('5091017137948', 'Ivan', 'David', '01-SEP-2015', '0742176201');  
insert into ELEVI values ('5120131451255', 'Ursu', 'Filip', '01-SEP-2018', '0726190221');  
insert into ELEVI values ('5130902228277', 'Lungu', 'Robert', '01-SEP-2019', '0743141321');  
insert into ELEVI values ('6080902174948', 'Dima', 'Antonia', '01-SEP-2014', '0740200625');  
insert into ELEVI values ('6121214351221', 'Curta', 'Anastasia', '01-SEP-2018', '0723557550');  
insert into ELEVI values ('6140821339777', 'Burlacu', 'Maria', '01-SEP-2020', '0729841235');  
insert into ELEVI values ('6100915027525', 'Mihalache', 'Cristina', '01-SEP-2016', '0747645884');  
insert into ELEVI values ('6110302445586', 'Sandu', 'Alexia', '01-SEP-2017', '0743044863');  
insert into ELEVI values ('6130409370399', 'Creanga', 'Elena', '01-SEP-2019', '0720042481');

1. Popularea tabelului SEDINTE

--TABEL SEDINTE  
insert into SEDINTE values (19, 'N', null, to\_date('14-NOV-2023 9:00:00', 'dd-mon-yyyy hh24:mi:ss'), 100, 25, '5010106415045');  
insert into SEDINTE values (29, 'Y', 'https://meet.google.com/ajn-agqy-zdr', to\_date('15-NOV-2023 11:00:00', 'dd-mon-yyyy hh24:mi:ss'), 80, null, '1910919412026');  
insert into SEDINTE values (39, 'N', null, to\_date('15-NOV-2023 11:00:00', 'dd-mon-yyyy hh24:mi:ss'), 90, 85, '6000620090130');  
insert into SEDINTE values (49, 'N', null, to\_date('15-NOV-2023 11:00:00', 'dd-mon-yyyy hh24:mi:ss'), 70, 95, '1810711417513');  
insert into SEDINTE values (59, 'N', null, to\_date('17-NOV-2023 14:00:00', 'dd-mon-yyyy hh24:mi:ss'), 110, 55, '1940131415026');  
insert into SEDINTE values (69, 'N', null, to\_date('18-NOV-2023 15:00:00', 'dd-mon-yyyy hh24:mi:ss'), 100, 155, '2960805527011');  
insert into SEDINTE values (79, 'N', null, to\_date('13-APR-2023 12:00:00', 'dd-mon-yyyy hh24:mi:ss'), 90, 75, '2840930240988');  
insert into SEDINTE values (89, 'N', null, to\_date('21-NOV-2022 17:00:00', 'dd-mon-yyyy hh24:mi:ss'), 80, 145, '2870906230977');  
insert into SEDINTE values (99, 'N', null, to\_date('10-DEC-2023 12:00:00', 'dd-mon-yyyy hh24:mi:ss'), 100, 45, '1910919412026');  
insert into SEDINTE values (109, 'N', null, to\_date('22-NOV-2023 10:00:00', 'dd-mon-yyyy hh24:mi:ss'), 110, 105, '2921024331691');  
insert into SEDINTE values (119, 'N', null, to\_date('29-NOV-2023 19:00:00', 'dd-mon-yyyy hh24:mi:ss'), 90, 25, '2760924018506');  
insert into SEDINTE values (129, 'Y', 'https://meet.google.com/zan-fzhm-cnm', to\_date('5-DEC-2022 13:00:00', 'dd-mon-yyyy hh24:mi:ss'), 70, null, '2760924018506');  
insert into SEDINTE values (139, 'Y', 'https://meet.google.com/zan-fzhm-cnm', to\_date('7-SEP-2023 8:00:00', 'dd-mon-yyyy hh24:mi:ss'), 60, null, '1960714417703');  
insert into SEDINTE values (149, 'Y', 'https://meet.google.com/sqm-hjhi-awv', to\_date('4-OCT-2023 10:00:00', 'dd-mon-yyyy hh24:mi:ss'), 75, null, '2890613280853');  
insert into SEDINTE values (159, 'Y', 'https://meet.google.com/iib-wpjw-asd', to\_date('7-OCT-2022 11:00:00', 'dd-mon-yyyy hh24:mi:ss'), 80, null, '2771031152107');  
insert into SEDINTE values (169, 'Y', 'https://meet.google.com/mvf-vtpy-ffm', to\_date('5-NOV-2021 14:00:00', 'dd-mon-yyyy hh24:mi:ss'), 65, null, '2870906230977');  
insert into SEDINTE values (179, 'Y', 'https://meet.google.com/yfx-danc-hkx', to\_date('10-MAY-2023 15:00:00', 'dd-mon-yyyy hh24:mi:ss'), 70, null, '1940131415026');  
insert into SEDINTE values (189, 'Y', 'https://meet.google.com/zej-wkzq-xyv', to\_date('20-DEC-2023 10:00:00', 'dd-mon-yyyy hh24:mi:ss'), 80, null, '1751024416058');  
insert into SEDINTE values (199, 'Y', 'https://meet.google.com/zej-wkzq-xyv', to\_date('20-DEC-2023 11:00:00', 'dd-mon-yyyy hh24:mi:ss'), 80, null, '1751024416058');

1. Popularea tabelului FACTURI

--TABEL FACTURI  
insert into FACTURI values (16, 19, '5150705035250', '14-NOV-2023', 'cash');  
insert into FACTURI values (26, 19, '6140821339777', '15-NOV-2023', 'card');  
insert into FACTURI values (36, 19, '6130409370399', '12-NOV-2023', 'card');  
insert into FACTURI values (46, 19, '5130902228277', '10-NOV-2023', 'cash');  
insert into FACTURI values (56, 189, '5091017137948', '21-DEC-2023', 'card');  
insert into FACTURI values (66, 199, '5091017137948', '21-DEC-2023', 'card');  
insert into FACTURI values (76, 89, '5130902228277', '20-NOV-2022', 'cash');  
insert into FACTURI values (86, 89, '6130409370399', '21-NOV-2022', 'cash');  
insert into FACTURI values (96, 109, '6100915027525', '30-NOV-2023', 'cash');  
insert into FACTURI values (106, 109, '6080902174948', '20-NOV-2023', 'cash');  
insert into FACTURI values (116, 109, '5091017137948', '22-NOV-2023', 'cash');  
insert into FACTURI values (126, 29, '6121214351221', '15-NOV-2023', 'card');  
insert into FACTURI values (136, 39, '5120131451255', '15-NOV-2023', 'card');  
insert into FACTURI values (146, 39, '6121214351221', '15-NOV-2023', 'cash');  
insert into FACTURI values (156, 49, '5100217419827', '10-NOV-2023', 'card');  
insert into FACTURI values (166, 59, '5110523111179', '16-NOV-2023', 'card');  
insert into FACTURI values (176, 59, '6110302445586', '18-NOV-2023', 'cash');  
insert into FACTURI values (186, 79, '5091017137948', '1-MAY-2023', 'cash');  
insert into FACTURI values (196, 99, '6080902174948', '21-NOV-2022', 'cash');  
insert into FACTURI values (206, 119, '5100217419827', '29-NOV-2023', 'cash');  
insert into FACTURI values (216, 129, '6100915027525', '5-DEC-2022', 'card');  
insert into FACTURI values (226, 139, '6121214351221', '7-SEP-2023', 'card');  
insert into FACTURI values (236, 139, '5120131451255', '8-SEP-2023', 'card');  
insert into FACTURI values (246, 149, '5150705035250', '4-OCT-2023', 'card');  
insert into FACTURI values (256, 149, '5130902228277', '4-OCT-2023', 'card');  
insert into FACTURI values (266, 149, '6140821339777', '3-OCT-2023', 'cash');  
insert into FACTURI values (276, 149, '6130409370399', '7-OCT-2023', 'cash');  
insert into FACTURI values (286, 159, '5100217419827', '8-OCT-2022', 'cash');  
insert into FACTURI values (296, 159, '5110523111179', '10-OCT-2022', 'cash');  
insert into FACTURI values (306, 159, '6100915027525', '6-OCT-2022', 'card');  
insert into FACTURI values (316, 159, '6110302445586', '5-OCT-2022', 'cash');  
insert into FACTURI values (326, 169, '6110302445586', '4-NOV-2021', 'card');  
insert into FACTURI values (336, 169, '5110523111179', '5-NOV-2021', 'card');  
insert into FACTURI values (346, 179, '6080902174948', '10-MAY-2023', 'card');  
insert into FACTURI values (356, 69, '5120131451255', '18-NOV-2023', 'cash');  
insert into FACTURI values (366, 69, '6121214351221', '18-NOV-2023', 'cash');  
insert into FACTURI values (376, 69, '5110523111179', '18-NOV-2023', 'cash');