Universitatea din București

Facultatea de Matematică și Informatică

Departamentul Calculatoare si Tehnologia Informației

**Proiect Baze de Date**

Coordonator științific: Student:

Vasile Silviu-Laurențiu Stan Robert-Ionuț

Universitatea din București

Facultatea de Matematică și Informatică

Departamentul Calculatoare si Tehnologia Informației

**Sistem de gestiune al unei firme IT**

Coordonator științific: Student:

Vasile Silviu-Laurențiu Stan Robert-Ionuț

**CUPRINS**

Prezentarea modelului (din lumea reală) și a regulilor acestuia.  **4**

Diagrama entitate-relație **5**

Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților **5**

1. Descrierea entităților, atributelor, cheilor și a relațiilor 5
2. Descrierea cardinalităților 9

Diagrama conceptuală **11**

Descrierea constrângerilor de integritate **11**

Schemele relaționale **14**

**Prezentarea modelului (din lumea reală) și a regulilor acestuia**

Gestionarea eficientă este esențială pentru succesul într-un mediu de afaceri competitiv. Prin organizare și planificare strategică, se optimizează procesele și se utilizează resursele cu înțelepciune. Acest lucru nu doar îmbunătățește eficiența, ci și stimulează creșterea și inovația.

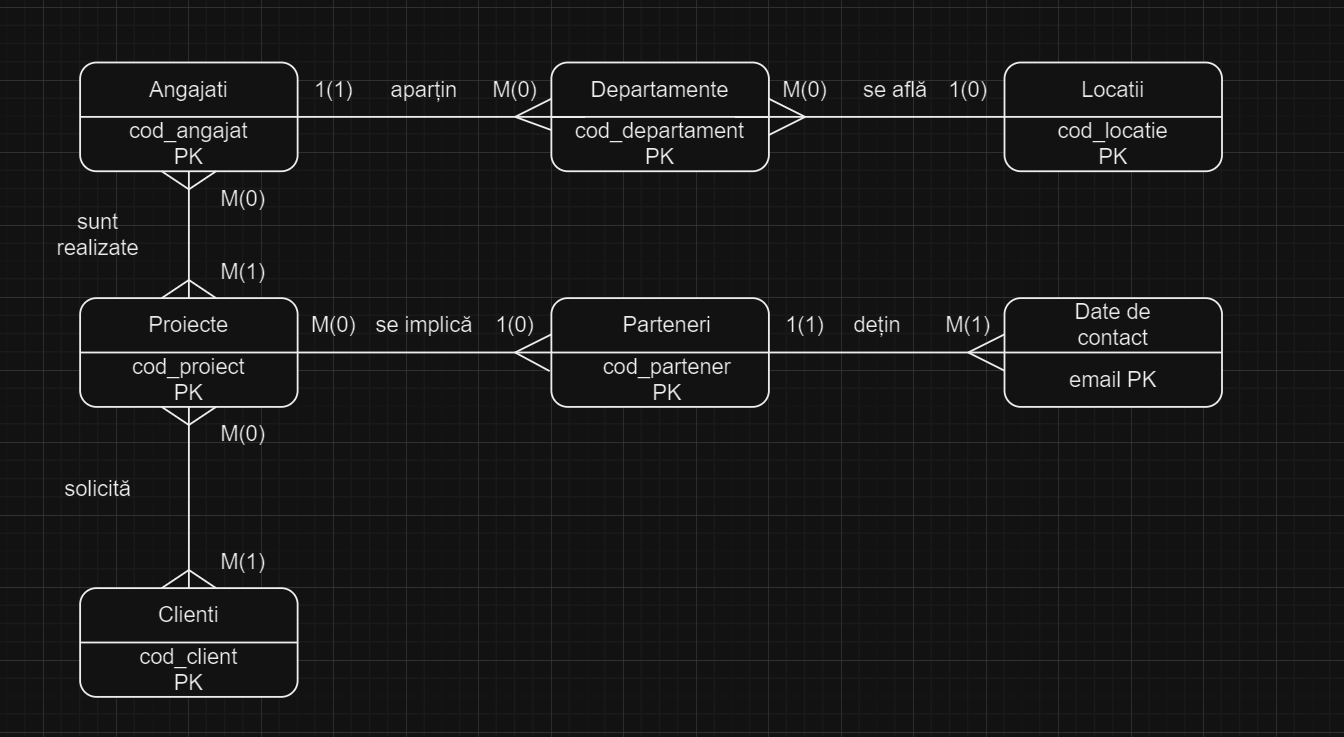
* **Locatii**: Reprezintă locațiile fizice ale organizației, inclusiv informații despre oraș, stradă și cod poștal. Fiecare locație are un cod unic (cod\_locatie) și este asociată cu cel puțin un departament.
* **Departamente**: Organizația este împărțită în mai multe departamente, fiecare asociat cu o anumită locație. Fiecare departament are un nume unic (nume\_departament) și este localizat într-o anumită locație.
* **Clienti**: Reprezintă persoane fizice sau companii care au interacțiuni cu organizația.

Un client este identificat printr-un cod unic (cod\_client) și poate fi o persoană fizică sau o companie.

* **Parteneri**: Partenerii sunt implicați în colaborări cu organizația. Fiecare partener are un cod unic (cod\_partener) și detine informații precum tipul și data începerii colaborării.
* **Date\_de\_contact**: Furnizează informații de contact pentru parteneri, inclusiv e-mailuri, numere de telefon și adrese Skype.
* **Angajati**: Reprezintă personalul organizației, inclusiv nume, prenume, data angajării și salariu. Fiecare angajat este identificat printr-un cod unic (cod\_angajat) și este asociat cu un departament.
* **Proiecte**: Descrie proiectele organizației, precum și stadiul actual al fiecărui proiect. Fiecare proiect are un cod unic (cod\_proiect) și este într-un anumit stadiu (stadiu\_proiect).
* **Proiecte\_Clienti**: Este o legătură între proiecte și clienți, indicând asocierile dintre ele.
* **Angajati\_Proiecte**: Este o legătură între angajați și proiecte, indicând participarea angajaților la diverse proiecte.

**Diagrama entitate-relație**

**(Sistem de gestiune al unei firme IT)**



**Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților**

**1. Descrierea entităților, atributelor, cheilor și a relațiilor**

**Tabelul Locatii:**

**Descriere entitate:**

Locatii reprezintă informații despre diverse locații în care se pot afla departamente.

**Descriere atribute:**

cod\_locatie (cheie primară): Identificator unic pentru locație.

oras: Numele orașului, validat să conțină doar litere.

strada: Numele străzii, validat să conțină doar litere.

cod\_postal: Codul poștal, validat să aibă exact 6 cifre.

**Chei:**

cod\_locatie (PK): Cheie primară pentru Locatii, identificator unic al locației.

Nu există chei străine definite în această tabelă.

**Tabelul Departamente:**

**Descriere entitate:**

Departamente reprezintă diferite unități organizaționale în cadrul unei companii.

**Descriere atribute:**

cod\_departament (cheie primară): Identificator unic pentru departament.

nume\_departament: Numele departamentului.

cod\_locatie: Cheie străină legată de cod\_locatie în Locatii, indicând locația departamentului.

**Chei:**

cod\_departament (PK): Cheie primară pentru Departamente, identificator unic al departamentului.

cod\_locatie din tabelul Departamente este cheie străină (FK) care referențiază cod\_locatie din tabelul Locatii.

**Tabelul Clienti:**

**Descriere entitate:**

Clienti reprezintă informații despre clienții organizației.

**Descriere atribute:**

cod\_client (cheie primară): Identificator unic pentru client.

nume\_client: Numele clientului, validat să aibă între 3 și 15 caractere.

prenume\_client: Prenumele clientului, validat să aibă între 3 și 30 de caractere.

tip\_client: Tipul clientului, fie "Persoana fizica" fie "Companie".

data\_ultimei\_achizitii: Data ultimei achiziții a clientului.

**Chei:**

cod\_client (PK): Cheie primară pentru Clienti, identificator unic al clientului.

Nu există chei străine definite în această tabelă.

**Tabelul Parteneri:**

**Descriere entitate:**

Parteneri reprezintă informații despre partenerii organizației.

**Descriere atribute:**

cod\_partener (cheie primară): Identificator unic pentru partener.

nume\_partener: Numele partenerului, validat să aibă între 3 și 30 de caractere.

tip\_partener: Tipul partenerului.

data\_incepere\_colaborare: Data începerii colaborării cu partenerul.

stare\_colaborare: Starea colaborării, fie "activa," "incheiata," fie "negociere."

comision\_acordat: Procentul de comision acordat partenerului.

**Chei:**

cod\_partener (PK): Cheie primară pentru Parteneri, identificator unic al partenerului.

Nu există chei străine definite în această tabelă.

**Tabelul Date\_de\_contact:**

**Descriere entitate:**

Date\_de\_contact reprezintă informații de contact asociate partenerilor.

**Descriere atribute:**

email (cheie primară): Adresa de email, validată să fie în formatul acceptat.

numar\_telefon: Numărul de telefon, validat să conțină doar cifre și să aibă exact 10 caractere.

cod\_partener: Cheie străină legată de cod\_partener în Parteneri, indicând partenerul asociat.

adresa\_skype: Adresa Skype, validată să aibă maxim 15 caractere.

C**hei:**

email (PK): Cheie primară pentru Date\_de\_contact, identificator unic al informațiilor de contact.

cod\_partener din tabelul Date\_de\_contact este o cheie străină (FK) care referențiază cod\_partener din tabelul Parteneri.

**Tabelul Angajati:**

**Descriere entitate:**

Angajati reprezintă informații despre angajații organizației.

**Descriere atribute:**

cod\_angajat (cheie primară): Identificator unic pentru angajat.

nume: Numele angajatului, validat să aibă între 3 și 15 caractere.

prenume: Prenumele angajatului, validat să aibă între 3 și 30 de caractere.

data\_angajarii: Data angajării angajatului.

salariu: Salariul angajatului, validat să fie mai mare sau egal cu 0.

cod\_departament: Cheie străină legată de cod\_departament în Departamente, indicând departamentul asociat.

**Chei:**

cod\_angajat (PK): Cheie primară pentru Angajati, identificator unic al angajatului.

cod\_departament din tabelul Angajati este o cheie străină (FK) care referențiază cod\_departament din tabelul Departamente.

**Tabelul Proiecte:**

**Descriere entitate:**

Proiecte reprezintă informații despre diferite proiecte în care organizația este implicată.

**Descriere atribute:**

cod\_proiect (cheie primară): Identificator unic pentru proiect.

nume\_proiect: Numele proiectului.

cod\_partener: Cheie străină legată de cod\_partener în Parteneri, indicând partenerul asociat.

stadiu\_proiect: Stadiul proiectului.

**Chei:**

cod\_proiect (PK): Cheie primară pentru Proiecte, identificator unic al proiectului.

cod\_partener din tabelul Proiecte este o cheie străină (FK) care referențiază cod\_partener din tabelul Parteneri.

**Tabelul Proiecte\_Clienti:**

**Descriere entitate:**

Proiecte\_Clienti reprezintă legătura dintre proiecte și clienți.

**Descriere atribute:**

cod\_proiect: Cheie străină legată de cod\_proiect în Proiecte, indicând proiectul asociat.

cod\_client: Cheie străină legată de cod\_client în Clienti, indicând clientul asociat.

**Chei:**

cod\_proiect, cod\_client (PK): Cheie primară compusă pentru Proiecte\_Clienti, indicând asocierea unică între proiect și client.

**Tabelul Angajati\_Proiecte:**

**Descriere entitate:**

Angajati\_Proiecte reprezintă legătura dintre angajați și proiecte.

**Descriere atribute:**

cod\_angajat: Cheie străină legată de cod\_angajat în Angajati, indicând angajatul asociat.

cod\_proiect: Cheie străină legată de cod\_proiect în Proiecte, indicând proiectul asociat.

**Chei:**

cod\_angajat, cod\_proiect (PK): Cheie primară compusă pentru Angajati\_Proiecte, indicând asocierea unică între angajat și proiect.

1. **Descrierea cardinalităților**

**Angajati - Departamente:**

Un angajat poate aparține doar unui departament

Nu există angajați fără departament

Pot exista mai multe departamente

Există departamente fără angajați

**Departamente - Locatii:**

Pot exista mai multe departamente

Există departamente fără locație

Un departament se poate afla doar într-o locație

Există locații fără departamente

**Angajati - Proiecte:**

Un angajat poate lucra la mai multe proiecte

Există angajați care nu lucrează la proiecte

Un proiect poate fi realizat de mai mulți angajați

Un proiect trebuie să fie realizat de cel puțin un angajat

**Proiecte - Parteneri:**

La un proiect se poate implica doar un partener

Există proiecte fără parteneri

Partenerii se pot implica în mai multe proiecte

Există parteneri fără proiecte

**Parteneri - Date de contact:**

Un partener poate avea mai multe date de contact

Un partener trebuie să aibă cel puțin o dată de contact

O dată de contact nu poate aparține mai multor parteneri

O dată de contact trebuie să aparțină cel puțin unui partener

**Proiecte - Clienti:**

Un proiect este solicitat de mai mulți clienți

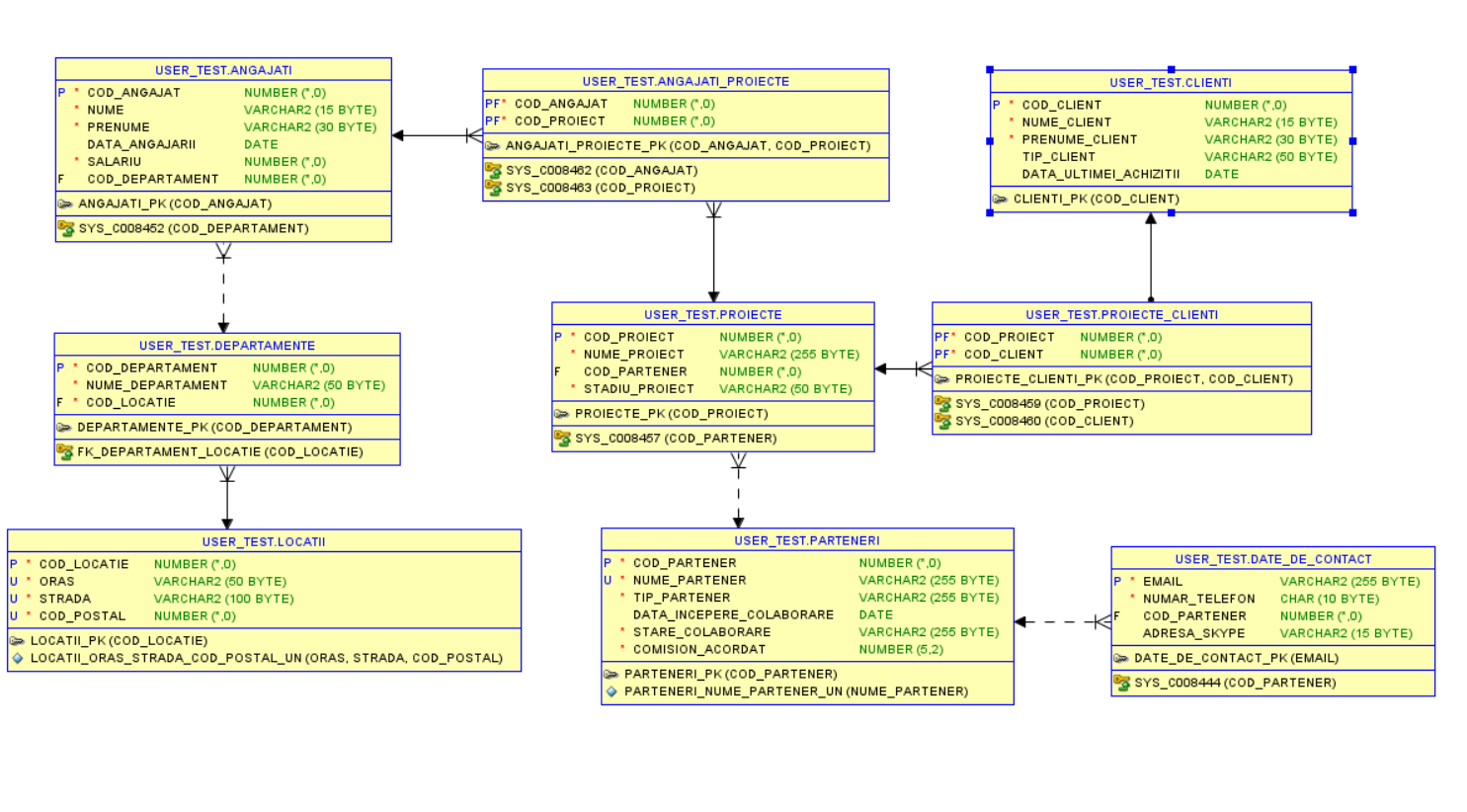
Există proiecte fără clienți

Un client poate solicita mai multe proiecte

Un client are cel puțin un proiect solicitat

**Diagrama conceptuală**

**(Sistem de gestiune al unei firme IT)**



**Descrierea constrângerilor de integritate**

**Locatii:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru cod\_locatie.

CHECK (REGEXP\_LIKE(oras, '^[a-zA-Z]+$')): Asigură că în coloana oras apar doar caractere alfabetice.

CHECK (REGEXP\_LIKE(strada, '^[a-zA-Z]+$')): Asigură că în coloana strada apar doar caractere alfabetice.

CHECK (LENGTH(TO\_CHAR(cod\_postal)) = 6): Asigură că elementul cod\_postal are exact 6 caractere.

UNIQUE (oras, strada, cod\_postal): Asigură că fiecare combinație de valori pentru coloanele oras, strada, și cod\_postal este unică.

**Departamente:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru cod\_departament.

FOREIGN KEY (cod\_locatie) REFERENCES Locatii(cod\_locatie): Creează o legătură între cod\_locatie din Departamente și cod\_locatie din Locatii.

**Clienti:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru cod\_client.

CHECK (LENGTH(nume\_client) >= 3 AND LENGTH(nume\_client) <= 15): Asigură că nume\_client are între 3 și 15 caractere.

CHECK (LENGTH(prenume\_client) >= 3 AND LENGTH(prenume\_client) <= 30): Asigură că prenume\_client are între 3 și 30 caractere.

VARCHAR(50) CHECK (tip\_client IN ('Persoana fizica', 'Companie')): Asigură ca tip\_client este o valoare dintre cele specificate.

**Parteneri:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru cod\_partener.

CHECK (LENGTH(nume\_partener) > 3 AND LENGTH(nume\_partener) < 30): Asigură că nume\_partener are între 3 și 30 caractere.

CHECK (stare\_colaborare IN ('activa', 'incheiata', 'negociere')): Asigură ca stare\_colaborare ia o valoare dintre cele specificate.

CHECK (comision\_acordat >= 0 AND comision\_acordat <= 100): Asigură că elementul comision\_acordat poate lua valori mai mari sau egale cu 0 și mai mici sau egale cu 100.

UNIQUE (nume\_partener): Asigură că nume\_partener este unic.

**Date\_de\_contact:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru email.

CHECK (REGEXP\_LIKE(email, '^[a-zA-Z0-9.\_%+-]+@(yahoo\.com|gmail\.com)$')): Asigură că email respectă un format specific.

CHECK (REGEXP\_LIKE(numar\_telefon, '^[0-9]{10}$')): Asigură că numar\_telefon să fie format din exact 10 cifre.

CHECK (LENGTH(adresa\_skype) <= 15): Asigură că adresa\_skype are maxim 15 caractere.

FOREIGN KEY (cod\_partener) REFERENCES Parteneri(cod\_partener): Creează o legătură între cod\_partener din Date\_de\_contact și cod\_partener din Parteneri.

**Angajati:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru cod\_angajat.

CHECK (LENGTH(nume) >= 3 AND LENGTH(nume) <= 15): Asigură că nume are între 3 și 15 caractere.

CHECK (LENGTH(prenume) >= 3 AND LENGTH(prenume) <= 30): Asigură că prenume are între 3 și 30 caractere.

CHECK (salariu >= 0): Asigură că salariu este mai mare sau egal cu 0

FOREIGN KEY (cod\_departament) REFERENCES Departamente(cod\_departament): Creează o legătură între cod\_departament din Angajati și cod\_departament din Departamente.

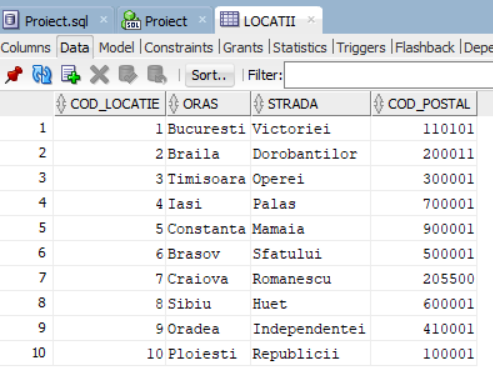
**Proiecte:**

PRIMARY KEY: Asigură că fiecare înregistrare din tabel are o valoare unică pentru cod\_proiect.

CHECK (stadiu\_proiect IN ('In desfasurare', 'Finalizat', 'Intarziat', 'Anulat', 'Planificat')): Asigură că stadiu\_proiect este una dintre valorile specificate.

FOREIGN KEY (cod\_partener) REFERENCES Parteneri(cod\_partener): Creează o legătură între cod\_partener din Proiecte și cod\_partener din Parteneri.

**Schemele relaționale**



-- Tabelul Locatii

CREATE TABLE Locatii (

    cod\_locatie INT PRIMARY KEY,

    oras VARCHAR(50) NOT NULL CHECK (REGEXP\_LIKE(oras, '^[a-zA-Z]+$')),

    strada VARCHAR(100) NOT NULL CHECK (REGEXP\_LIKE(strada, '^[a-zA-Z]+$')),

    cod\_postal cod\_postal INT NOT NULL CHECK (LENGTH(TO\_CHAR(cod\_postal)) = 6),

    UNIQUE (oras, strada, cod\_postal)

);

-- Inserare date în Locatii

INSERT INTO Locatii (cod\_locatie, oras, strada, cod\_postal) VALUES (1, 'Bucuresti', 'Victoriei', 110101);

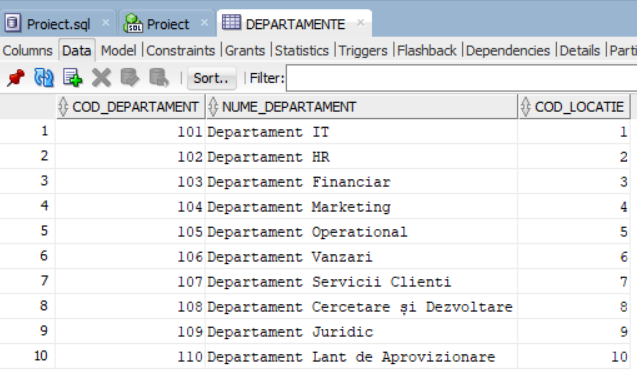
INSERT INTO Locatii (cod\_locatie, oras, strada, cod\_postal) VALUES (2, 'Braila', 'Dorobantilor', 200011);

INSERT INTO Locatii (cod\_locatie, oras, strada, cod\_postal) VALUES (3, 'Timisoara', 'Operei', 300001);

INSERT INTO Locatii (cod\_locatie, oras, strada, cod\_postal) VALUES (4, 'Iasi', 'Palas', 700001);

INSERT INTO Locatii (cod\_locatie, oras, strada, cod\_postal) VALUES (5, 'Constanta', 'Mamaia', 900001);

INSERT INTO Locatii (cod\_locatie, oras, strada, cod\_postal) VALUES (6, 'Brasov', 'Sfatului', 500001);



-- Tabelul Departamente

CREATE TABLE Departamente (

    cod\_departament INT PRIMARY KEY,

    nume\_departament VARCHAR(50) NOT NULL,

    cod\_locatie INT NOT NULL,

    CONSTRAINT fk\_departament\_locatie FOREIGN KEY (cod\_locatie) REFERENCES Locatii(cod\_locatie)

);

-- Inserare date în Departamente

INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(101, 'Departament IT', 1);

INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(102, 'Departament HR', 2);

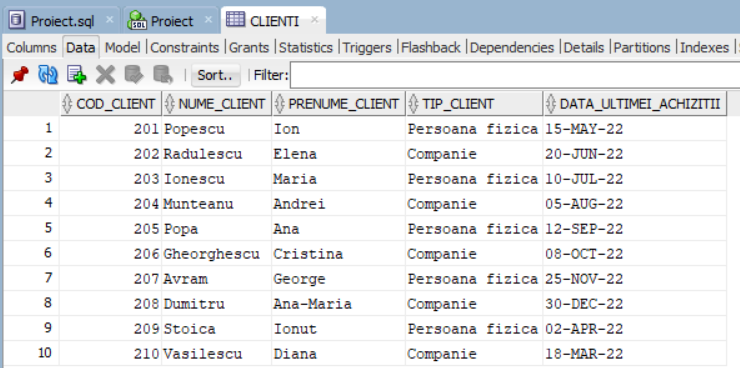
INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(103, 'Departament Financiar', 3);

INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(104, 'Departament Marketing', 4);

INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(105, 'Departament Operational', 5);

INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(106, 'Departament Vanzari', 6);

INSERT INTO Departamente (cod\_departament, nume\_departament, cod\_locatie) VALUES(107, 'Departament Servicii Clienti', 7);



-- Tabelul Clienti

CREATE TABLE Clienti (

    cod\_client INT PRIMARY KEY,

    nume\_client VARCHAR(15) NOT NULL CHECK (LENGTH(nume\_client) >= 3 AND LENGTH(nume\_client) <= 15),

    prenume\_client VARCHAR(30) NOT NULL CHECK (LENGTH(prenume\_client) >= 3 AND LENGTH(prenume\_client) <= 30),

    tip\_client VARCHAR(50) CHECK (tip\_client IN ('Persoana fizica', 'Companie')),

    data\_ultimei\_achizitii DATE

);

-- Inserare date în Clienti

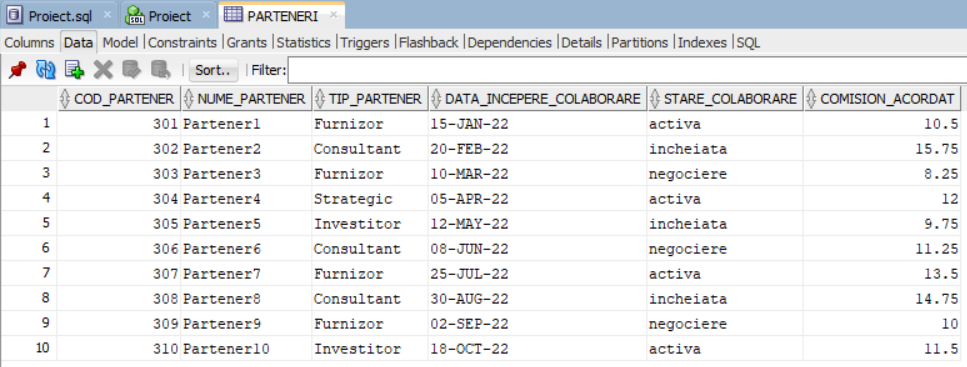
INSERT INTO Clienti (cod\_client, nume\_client, prenume\_client, tip\_client, data\_ultimei\_achizitii) VALUES(201, 'Popescu', 'Ion', 'Persoana fizica', TO\_DATE('2022-05-15', 'YYYY-MM-DD'));

INSERT INTO Clienti (cod\_client, nume\_client, prenume\_client, tip\_client, data\_ultimei\_achizitii) VALUES(202, 'Radulescu', 'Elena', 'Companie', TO\_DATE('2022-06-20', 'YYYY-MM-DD'));

INSERT INTO Clienti (cod\_client, nume\_client, prenume\_client, tip\_client, data\_ultimei\_achizitii) VALUES(203, 'Ionescu', 'Maria', 'Persoana fizica', TO\_DATE('2022-07-10', 'YYYY-MM-DD'));

INSERT INTO Clienti (cod\_client, nume\_client, prenume\_client, tip\_client, data\_ultimei\_achizitii) VALUES(204, 'Munteanu', 'Andrei', 'Companie', TO\_DATE('2022-08-05', 'YYYY-MM-DD'));

INSERT INTO Clienti (cod\_client, nume\_client, prenume\_client, tip\_client, data\_ultimei\_achizitii) VALUES(205, 'Popa', 'Ana', 'Persoana fizica', TO\_DATE('2022-09-12', 'YYYY-MM-DD'));



-- Tabelul Parteneri

CREATE TABLE Parteneri (

    cod\_partener INT PRIMARY KEY,

    nume\_partener VARCHAR(255) NOT NULL CHECK (LENGTH(nume\_partener) > 3 AND LENGTH(nume\_partener) < 30),

    tip\_partener VARCHAR(255) NOT NULL,

    data\_incepere\_colaborare DATE,

    stare\_colaborare VARCHAR(255) NOT NULL CHECK (stare\_colaborare IN ('activa', 'incheiata', 'negociere')),

    comision\_acordat DECIMAL(5,2) NOT NULL CHECK (comision\_acordat >= 0 AND comision\_acordat <= 100),

    UNIQUE (nume\_partener)

);

-- Inserare date în Parteneri

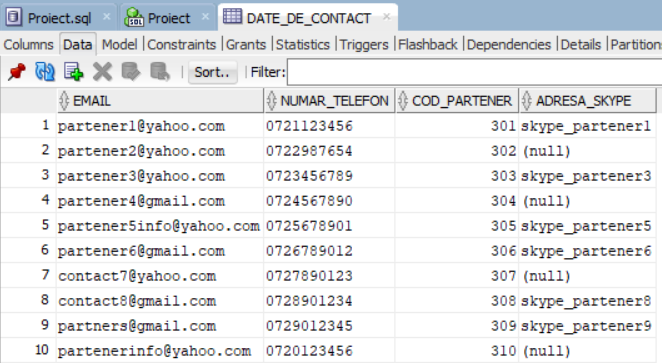
INSERT INTO Parteneri (cod\_partener, nume\_partener, tip\_partener, data\_incepere\_colaborare, stare\_colaborare, comision\_acordat) VALUES (301, 'Partener1', 'Furnizor', TO\_DATE('2022-01-15', 'YYYY-MM-DD'), 'activa', 10.50);

INSERT INTO Parteneri (cod\_partener, nume\_partener, tip\_partener, data\_incepere\_colaborare, stare\_colaborare, comision\_acordat) VALUES (302, 'Partener2', 'Consultant', TO\_DATE('2022-02-20', 'YYYY-MM-DD'), 'incheiata', 15.75);

INSERT INTO Parteneri (cod\_partener, nume\_partener, tip\_partener, data\_incepere\_colaborare, stare\_colaborare, comision\_acordat) VALUES (303, 'Partener3', 'Furnizor', TO\_DATE('2022-03-10', 'YYYY-MM-DD'), 'negociere', 8.25);

INSERT INTO Parteneri (cod\_partener, nume\_partener, tip\_partener, data\_incepere\_colaborare, stare\_colaborare, comision\_acordat) VALUES (304, 'Partener4', 'Strategic', TO\_DATE('2022-04-05', 'YYYY-MM-DD'), 'activa', 12.00);

INSERT INTO Parteneri (cod\_partener, nume\_partener, tip\_partener, data\_incepere\_colaborare, stare\_colaborare, comision\_acordat) VALUES (305, 'Partener5', 'Investitor', TO\_DATE('2022-05-12', 'YYYY-MM-DD'), 'incheiata', 9.75);



-- Tabelul Date\_de\_contact

CREATE TABLE Date\_de\_contact (

    email VARCHAR(255) PRIMARY KEY CHECK (REGEXP\_LIKE(email, '^[a-zA-Z0-9.\_%+-]+@(yahoo\.com|gmail\.com)$')),

    numar\_telefon CHAR(10) NOT NULL CHECK (REGEXP\_LIKE(numar\_telefon, '^[0-9]{10}$')),

    cod\_partener INT,

    adresa\_skype VARCHAR(15) CHECK (LENGTH(adresa\_skype) <= 15),

    FOREIGN KEY (cod\_partener) REFERENCES Parteneri(cod\_partener)

);

-- Inserare date în Date\_de\_contact

INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('partener1@yahoo.com', '0721123456', 301, 'skype\_partener1');

INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('partener2@yahoo.com', '0722987654', 302, NULL);

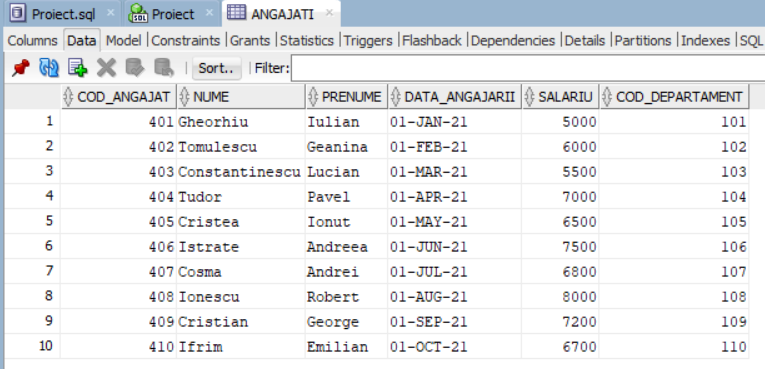
INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('partener3@yahoo.com', '0723456789', 303, 'skype\_partener3');

INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('partener4@gmail.com', '0724567890', 304, NULL);

INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('partener5info@yahoo.com', '0725678901', 305, 'skype\_partener5');

INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('partener6@gmail.com', '0726789012', 306, 'skype\_partener6');

INSERT INTO Date\_de\_contact (email, numar\_telefon, cod\_partener, adresa\_skype) VALUES ('contact7@yahoo.com', '0727890123', 307, NULL);



-- Tabelul Angajati

CREATE TABLE Angajati (

    cod\_angajat INT PRIMARY KEY,

    nume VARCHAR(15) NOT NULL CHECK (LENGTH(nume) >= 3 AND LENGTH(nume) <= 15),

    prenume VARCHAR(30) NOT NULL CHECK (LENGTH(prenume) >= 3 AND LENGTH(prenume) <= 30),

    data\_angajarii DATE,

    salariu DECIMAL NOT NULL CHECK (salariu >= 0),

    cod\_departament INT,

    FOREIGN KEY (cod\_departament) REFERENCES Departamente(cod\_departament)

);

-- Inserare date în Angajati

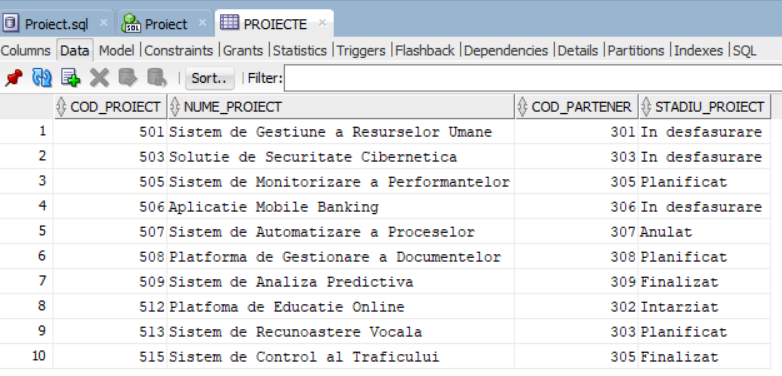
INSERT INTO Angajati (cod\_angajat, nume, prenume, data\_angajarii, salariu, cod\_departament) VALUES (401, 'Gheorhiu', 'Iulian', TO\_DATE('2021-01-01', 'YYYY-MM-DD'), 5000.0, 101);

INSERT INTO Angajati (cod\_angajat, nume, prenume, data\_angajarii, salariu, cod\_departament) VALUES (402, 'Tomulescu', 'Geanina', TO\_DATE('2021-02-01', 'YYYY-MM-DD'), 6000.0, 102);

INSERT INTO Angajati (cod\_angajat, nume, prenume, data\_angajarii, salariu, cod\_departament) VALUES (403, 'Constantinescu', 'Lucian', TO\_DATE('2021-03-01', 'YYYY-MM-DD'), 5500.0, 103);

INSERT INTO Angajati (cod\_angajat, nume, prenume, data\_angajarii, salariu, cod\_departament) VALUES (404, 'Tudor', 'Pavel', TO\_DATE('2021-04-01', 'YYYY-MM-DD'), 7000.0, 104);

INSERT INTO Angajati (cod\_angajat, nume, prenume, data\_angajarii, salariu, cod\_departament) VALUES (405, 'Cristea', 'Ionut', TO\_DATE('2021-05-01', 'YYYY-MM-DD'), 6500.0, 105);



-- Tabelul Proiecte

CREATE TABLE Proiecte (

    cod\_proiect INT PRIMARY KEY,

    nume\_proiect VARCHAR(255) NOT NULL,

    cod\_partener INT,

    stadiu\_proiect VARCHAR(50) NOT NULL CHECK (stadiu\_proiect IN ('In desfasurare', 'Finalizat', 'Intarziat', 'Anulat', 'Planificat')),

    FOREIGN KEY (cod\_partener) REFERENCES Parteneri(cod\_partener)

);

-- Inserare date în Proiecte

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(501, 'Sistem de Gestiune a Resurselor Umane', 301, 'In desfasurare');

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(503, 'Solutie de Securitate Cibernetica', 303, 'In desfasurare');

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(505, 'Sistem de Monitorizare a Performantelor', 305, 'Planificat');

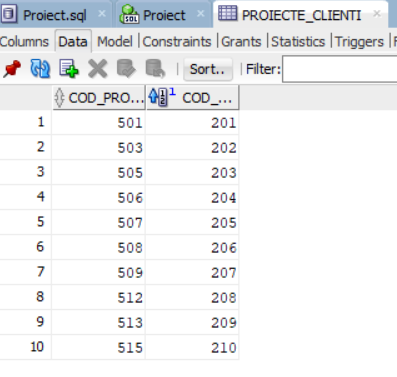
INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(506, 'Aplicatie Mobile Banking', 306, 'In desfasurare');

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(507, 'Sistem de Automatizare a Proceselor', 307, 'Anulat');

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(508, 'Platforma de Gestionare a Documentelor', 308, 'Planificat');

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(509, 'Sistem de Analiza Predictiva', 309, 'Finalizat');

INSERT INTO Proiecte (cod\_proiect, nume\_proiect, cod\_partener, stadiu\_proiect) VALUES(512, 'Platfoma de Educatie Online', 302, 'Intarziat');



-- Tabelul Proiecte\_Clienti

CREATE TABLE Proiecte\_Clienti (

    cod\_proiect INT,

    cod\_client INT,

    PRIMARY KEY (cod\_proiect, cod\_client),

    FOREIGN KEY (cod\_proiect) REFERENCES Proiecte(cod\_proiect),

    FOREIGN KEY (cod\_client) REFERENCES Clienti(cod\_client)

);

-- Inserare date în Proiecte\_Clienti

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(501,201);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(503,202);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(505,203);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(506,204);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(507,205);

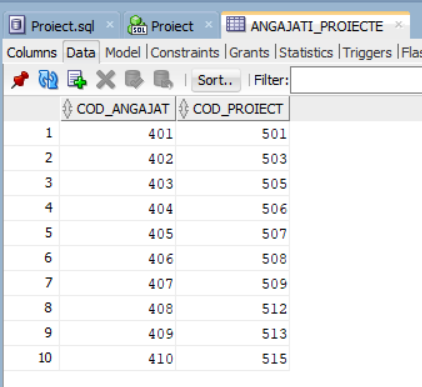
INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(508,206);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(509,207);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(512,208);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(513,209);

INSERT INTO Proiecte\_Clienti (cod\_proiect, cod\_client) VALUES(515,210);



-- Tabelul Angajati\_Proiecte

CREATE TABLE Angajati\_Proiecte (

    cod\_angajat INT,

    cod\_proiect INT,

    PRIMARY KEY (cod\_angajat, cod\_proiect),

    FOREIGN KEY (cod\_angajat) REFERENCES Angajati(cod\_angajat),

    FOREIGN KEY (cod\_proiect) REFERENCES Proiecte(cod\_proiect)

);

-- Inserare date în Angajati\_Proiecte

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(401, 501);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(402, 503);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(403, 505);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(404, 506);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(405, 507);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(406, 508);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(407, 509);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(408, 512);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(409, 513);

INSERT INTO Angajati\_Proiecte (cod\_angajat, cod\_proiect) VALUES(410, 515);