

Proiect la Baze de Date

Tema: Gestionarea unei firme de IT

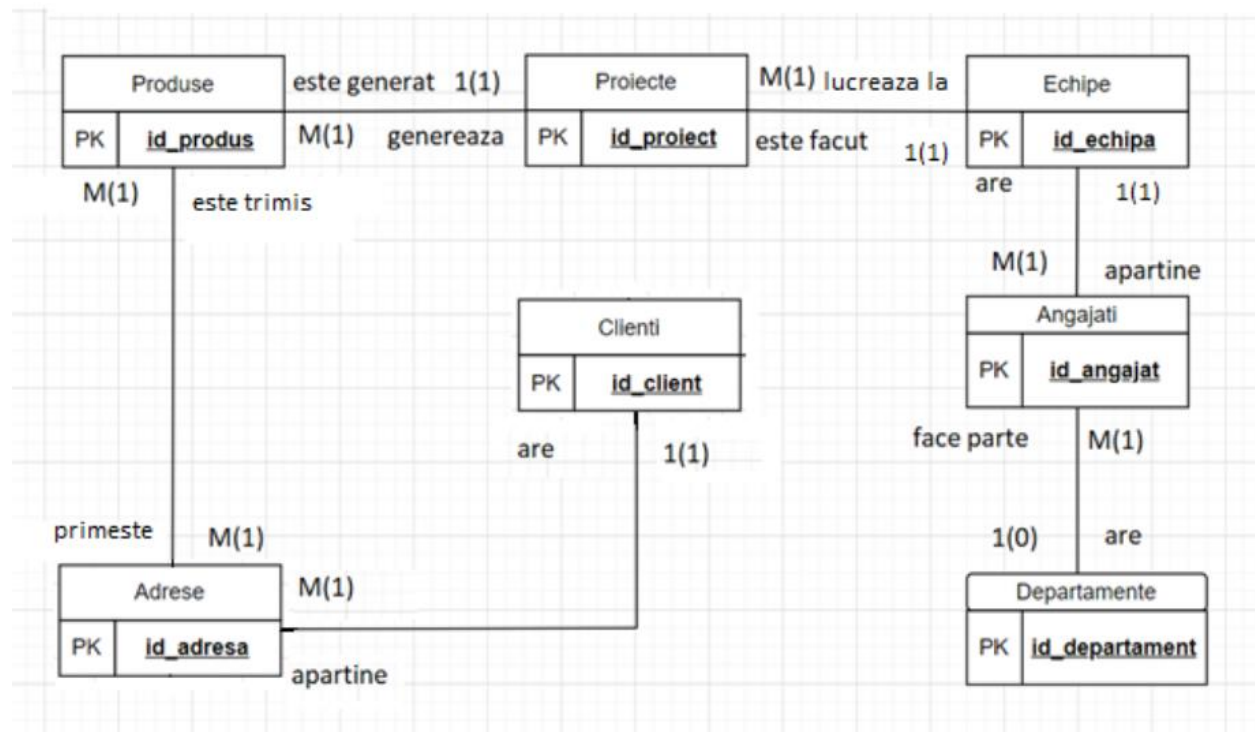
Profesor coordonator:
Vasile Silviu-Laurențiu

Student:
Gava Bogdan

1. Prezentarea modelului și a regulilor acestuia:

Aplicatia pe care doresc sa o implementez, in viziunea mea, functioneaza in felul urmator: Angajatii unei firme din segmentul tehnologiei informatiei fac parte din departamente in functie de domeniul lor de expertiza si vor fi grupati pe echipe. Fiecare echipa va lucra la mai multe proiecte pe parcursul timpului, de pe urma carora vor rezulta aplicatii software. Clientii vor face o comanda pentru un produs sau mai multe, care va fi livrata la adresa specificata de client.

2. Diagrama Entitati-Relatii:



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

1. Departamente (id_departament (PK), nume_departament)

Descriere entitate:

- Retine informatii privind departamentele.

Descriere atribut:

- nume_departament – departamentele vor dispune de un nume ce va descrie domeniul lor.

Chei:

- id_departament (PK) – identificator unic pentru Departamente.

Relatii:

- Cu entitatea Angajati: un department are mai multi angajati. Un angajat apartine unui singur department.

2. Echipa (id echipa (PK), nume)

Descriere entitate:

Retine numele si identificatorul unic pentru fiecare echipa.

Descriere atribut:

- nume – echipele vor avea un nume descriptiv activitatii membrilor.

Chei:

- id echipa (PK) - identificator unic pentru Echipa.

Relatii:

- Cu entitatea Angajati: o echipa are unul sau mai multi angajati. Un angajat apartine unei echipe.
- Cu entitatea Proiecte: O echipa intocmeste unul sau mai multe proiecte. Un proiect este facut de o echipa.

3. Clienti (id_client (PK), nume, prenume, numar_telefon, email)

Descriere entitate:

Are informatii despre persoanele care achizitioneaza produse sau servicii de la aceasta firma.

Descriere attribute:

- nume – numele de familie al clientului.
- prenume – prenumele clientului.
- numar_telefon – numarul de telefon pentru contactarea clientului.
- email – email-ul pentru a contacta clientul.

Chei:

- id_client (PK) – identificator unic pentru client

Relatii:

- Cu entitatea Adrese: Un client are una sau mai multe adrese. O adresa apartine unui client
4. Adrese (id_adresa (PK), judet, localitate, strada, numar, bloc, scara, apartament)

Descriere entitate:

Retine detalii privind adresa clientilor.

Descriere atribut:

- Judet – judetul unde se va livra comanda
- Localitate – localitatea unde se va livra comanda
- Strada – strada unde se va livra comanda
- Numar – numarul strazii
- Bloc – blocul unde se va livra comanda
- Scara – scara unde se va livra comanda
- Apartament – apartamentul unde se va livra comanda

Chei:

- Id_adresa (PK) – identificatorul unic pentru adresa
- Id_client (FK) – formeaza relatia cu Clienti

Relatii:

- Cu relatia Clienti: O adresa apartine unui client, iar un client are una sau mai multe adrese.
- Cu relatia Produse: intre Adrese si Produse se afla relatia many-to-many, asa ca am creat un tabel nou numit Comenzi pentru rezolvare.

5. Angajati (id_angajat (PK), id_departament (FK), id echipa (FK), nume, prenume, numar_telefon, data_angajarii, email, adresa, salariu)

Descriere entitate:

Retine datele angajatilor firmei.

Descriere atribute:

- nume – numele de familie al angajatului
- prenume – prenumele angajatului
- numar_telefon – numarul de telefon al angajatului
- email – email-ul personal al angajatului
- data_angajarii – data in care s-a facut angajarea fiecaruia
- data_nasterii – data in care s-a nascut angajatul
- salariu – salariul net al angajatului

Chei:

- id_angajat (PK) – identificator unic pentru Angajati
- id_departament (FK) – formeaza relatia cu Departamente
- id echipa (FK) – formeaza relatia cu Echipe

Relatii:

- Cu entitatea Echipe: un angajat apartine unei echipe. O echipa are unul sau mai multi angajati
- Cu entitatea Departamente: un angajat face parte dintr-un department. Un department are unul sau mai multi angajati

6. Proiecte (id_proiect (PK), id echipa (FK), nume, data_inceperii, data_finalizarii)

Descriere entitate:

Retine informatii legate de proiecte, mai exact cum se numeste, cand a fost inceput si cand va fi finalizat.

Descriere atribute:

- nume – numele proiectului
- data_inceperii – cand a fost inceput sa se lucreze la proiect
- data_finalizarii – cand proiectul a fost terminat

Chei:

- id_proiect (PK) – indentificator unic pentru Proiecte
- id echipa (FK) – formeaza relatia cu Echipe

Relatii:

- Cu entitatea Produse: Un proiect genereaza unul sau mai multe produse. Un produs este generat de un proiect.
- Cu entitatea Echipe: Un proiect este facut de o echipa. O echipa lucreaza la unul sau mai multe proiecte.

7. Produse (id_produs (PK), nume, pret)

Descriere entitate:

Retine informatii legate de produsele firmei

Descriere atribute:

- nume – denumirea produsului
- pret – pretul produsul

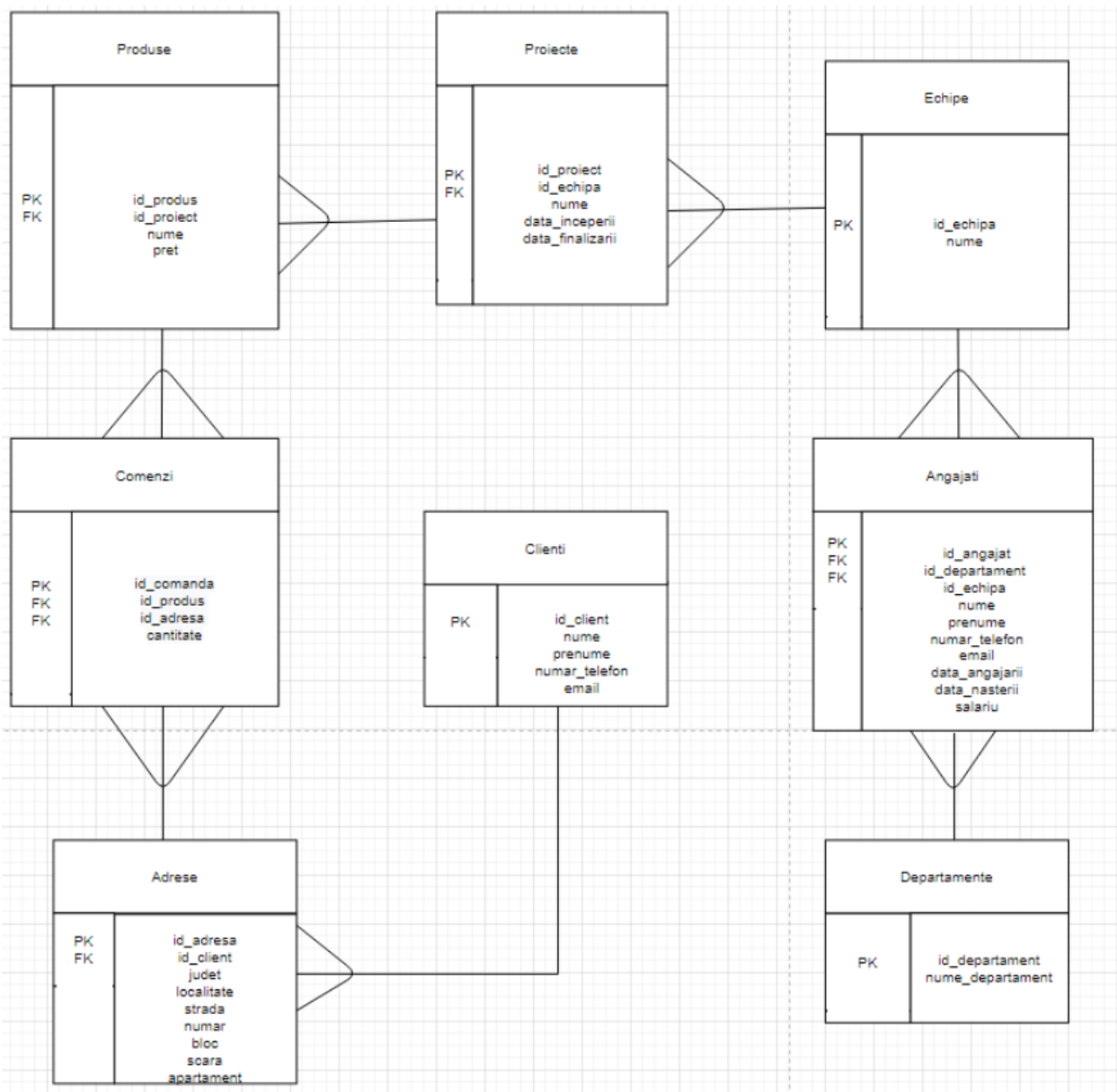
Chei:

- id_produs (PK) – identificator unic pentru Produse
- id_proiect (FK) – formeaza relatia cu Proiecte

Relatii:

- Cu entitatea Proiecte: Un produs este generat de un proiect. Un proiect genereaza mai multe produse.
- Cu entitatea Adrese: intre Produse si Adrese se afla relatia many-to-many, asa ca am creat un tabel nou numit Comenzi pentru rezolvare.

Diagrama conceptuala:



Descrierea constrangerilor de integritate

Departamente

- id_departament este PK.
- nume_departament trebuie sa aiba lungime de cel putin 1 caracter.

Echipe

- id_echipe este PK
- nume_echipe trebuie sa aiba lungime de cel putin 1 caracter

Clienti

- id_client este PK
- nume este diferit de NULL.
- prenume este diferit de NULL.
- email este UNIQUE si diferit de NULL.
- numar_telefon este UNIQUE si diferit de NULL.

Adrese

- id_adresa este PK.
- judet este diferit de NULL.
- localitate este diferit de NULL.
- numar este diferit de NULL.
- id_client este FK si NOT NULL.

Angajati

- id_angajati este PK
- nume este diferit de NULL.
- prenume este diferit de NULL.
- data_angajarii este diferit de NULL.
- data_nasterii este diferit de NULL.
- salariu este diferit de NULL.
- email este UNIQUE si diferit de NULL.
- numar_telefon este UNIQUE si diferit de NULL.
- id_departament este FK.
- id echipa este FK si NOT NULL.

Proiecte

- id_proiect va fi PK.
- nume trebuie sa aiba lungime de cel putin 1 caracter.
- data_inceperii este diferit de NULL.
- data_finalizarii este diferit de NULL.
- id echipa este FK si NOT NULL

Produse

- id_produs va fi PK
- nume este diferit de NULL.
- pret este diferit de NULL.
- id_proiect este FK si NOT NULL.

Comenzi

- id_comanda va fi PK
- cantitate este diferit de NULL.
- id_produs este FK si NOT NULL
- id_adresa este FK si NOT NULL

Scheme relationale

La stergerea unui department din Departamente, toti angajatii care fac parte din acel department vor fi stersi (ON DELETE CASCADE).

La stergerea unei echipe din Echipe, toti angajatii care fac parte din acea echipa vor fi stersi (ON DELETE CASCADE).

La stergerea unei echipe din Echipe, toate proiectele din Proiecte la care lucreaza acea echipa vor fi sterse (ON DELETE CASCADE).

La stergerea unui proiect din Proiecte, toate produsele rezultate din acele proiecte vor fi sterse (ON DELETE CASCADE).

La stergerea unui produs din Produse, toate comenzile care au acel produs vor fi sterse (ON DELETE CASCADE).

La stergerea unei adrese din Adrese, toate comenzile care au acea adresa vor fi sterse (ON DELETE CASCADE).

La stergerea unui client din Clienti, toate adresele care apartin acelui client vor fi sterse (ON DELETE CASCADE).

II. Implementarea intr-un sistem de gestiune a bazelor de date.

Pentru crearea tabelor si introducerea datelor am folosit Microsoft SQL Server.

a) Crearea tabelor (inclusiv a constrangerilor):

Departamente

```
CREATE TABLE Departamente
(
    id_departament INT primary key,
    nume_departament VARCHAR(30) constraint ch_nume_departament check
(len(nume_departament)>1)
);
```

Echipe

```
CREATE TABLE Echipe
(
    id echipa INT primary key,
    nume VARCHAR(30) constraint ch_nume check (len(nume)>=1)
);
```

Cienti

```
CREATE TABLE Cienti
(
    id_client INT constraint pk_client primary key,
    nume VARCHAR(30) NOT NULL,
    prenume VARCHAR(30) NOT NULL,
    numar_telefon VARCHAR(11) NOT NULL unique,
    email VARCHAR(50) NOT NULL unique,
);
```

Adrese

```
CREATE TABLE Adrese
(
    id_adresa INT primary key,
    judet VARCHAR(30) NOT NULL,
    localitate VARCHAR(50) NOT NULL,
    strada VARCHAR(40),
    numar INT not null,
    bloc VARCHAR(10),
    scara VARCHAR(10),
    apartament INT,
    id_client INT NOT NULL,
    constraint fk_id_client FOREIGN KEY (id_client) REFERENCES
Cienti(id_client) on delete cascade
);
```

Angajati

```
CREATE TABLE Angajati
(
    id_angajat INT primary key,
    nume VARCHAR(30) NOT NULL,
    prenume VARCHAR(30) NOT NULL,
    numar_telefon VARCHAR(11) NOT NULL unique,
    email VARCHAR(50) NOT NULL unique,
    data_angajarii DATE NOT NULL,
    data_nasterii DATE NOT NULL,
    salariu INT NOT NULL,
    id_departament INT,
    id echipa INT NOT NULL,
    constraint fk1_departament FOREIGN KEY (id_departament) REFERENCES
Departamente(id_departament) on delete cascade,
    constraint fk2 echipa FOREIGN KEY (id echipa) REFERENCES
Echipe(id echipa) on delete cascade
);
```

Proiecte

```
CREATE TABLE Proiecte
(
    id_proiect INT constraint pk_proiect primary key,
    data_inceperii DATE NOT NULL,
    data_finalizarii DATE NOT NULL,
    nume_proiect VARCHAR(30) constraint ch_nume_proiect check (len(nume_proiect)>=1),
    id echipa INT NOT NULL,
    constraint fk_id echipa FOREIGN KEY (id echipa) REFERENCES Echipe(id echipa) on delete
cascade
);
alter table Proiecte
    add constraint check_dates check (data_inceperii < data_finalizarii);
```

Produse

```
CREATE TABLE Produse
(
    id_produs INT primary key,
    nume VARCHAR(30) NOT NULL,
    pret INT NOT NULL,
    id_proiect INT NOT NULL,
```

```

    FOREIGN KEY (id_proiect) REFERENCES Proiecte(id_proiect) on delete
    cascade
);

```

Comenzi

```

CREATE TABLE Comenzi
(
    cantitate INT NOT NULL,
    id_comanda INT constraint pk_constraint primary key,
    id_produc INT NOT NULL,
    id_adresa INT NOT NULL,
    FOREIGN KEY (id_produc) REFERENCES Produse(id_produc) on delete
    cascade,
    FOREIGN KEY (id_adresa) REFERENCES Adrese(id_adresa) on delete
    cascade
);

```

b) Introdúcere date

```

insert into Departamente values (1, 'Resurse Umane')
insert into Departamente values (2, 'Finante')
insert into Departamente values (3, 'Dezvoltare')

insert into Echipe values (1, 'Fullstack1')
insert into Echipe values (2, 'Fullstack2')
insert into Echipe values (3, 'Back-end')
insert into Echipe values (4, 'Back-end2')

insert into Clienti values (1, 'Popescu', 'Radu', '0755746112', 'pradu@yahoo.com')
insert into Clienti values (2, 'Oprescu', 'Alexandru', '0755776112', 'alexo@yahoo.com')
insert into Clienti values (3, 'Ionescu', 'Ivan', '0755746512', 'ioni@yahoo.com')

insert into Adrese values (1, 'Arad', 'Nadlac', 'Mihai Eminescu', 2, 'F5', 'A', 12, 1)
insert into Adrese values (2, 'Arges', 'Pitesti', 'I. C. Bratianu', 5, 'A5', 'A', 2, 1)
insert into Adrese values (3, 'Arges', 'Campulung', 'Negru Voda', 66, NULL, NULL, NULL, 2)
insert into Adrese values (4, 'Prahova', 'Campina', 'Primaverii', 38, NULL, NULL, NULL, 2)
insert into Adrese values (5, 'Vrancea', 'Focsani', 'Nicolae Iorga', 21, 'B2', 'D', 5, 3)
insert into Adrese values (6, 'Bacau', 'Bacau', 'Dumbravei', 5, 'A1', 'C', 8, 3)

insert into Angajati values (1, 'Popescu', 'Andrei', '0768465988', 'ap@yahoo.com', '12 APR 1990', '20
DEC 2018', 6000, 1, NULL)
insert into Angajati values (2, 'Popa', 'Alexandru', '0768325948', 'palex@yahoo.com', '10 AUG
1995', '2 JUN 2019', 3000, 3, 1)
insert into Angajati values (3, 'Alexandrescu', 'Theodora', '0768465945', 'ath@yahoo.com', '12 MAR
1986', '25 FEB 2013', 12000, 3, 2)
insert into Angajati values (4, 'Cristescu', 'Marian', '0768465345', 'cma@yahoo.com', '17 APR
1993', '20 NOV 2018', 5000, 3, 3)
insert into Angajati values (5, 'Serban', 'Bianca', '0768465789', 'sbi@yahoo.com', '20 MAY 1998', '2
OCT 2017', 6300, 3, 4)

```

```

insert into Angajati values (6, 'Nan', 'Marius', '0768465357', 'nmar@yahoo.com', '31 AUG 1998', '5
DEC 2019', 6700, 3, 1)
insert into Angajati values (7, 'Achimescu', 'Ioana', '0768265935', 'aio@yahoo.com', '1 APR 1990', '8
SEP 2018', 8000, 3, 3)
insert into Angajati values (8, 'Tuta', 'Alin', '0762346588', 'tali@yahoo.com', '2 MAR 1982', '10 AUG
2011', 15000, NULL, 2)
insert into Angajati values (9, 'Tuta', 'Roxana', '0768463488', 'trox@yahoo.com', '12 JUL 1993', '3
JAN 2017', 5000, 3, 4)
insert into Angajati values (10, 'Vladimirescu', 'Bogdan', '0768237988', 'bogv@yahoo.com', '13 JAN
1989', '13 JUN 2015', 8600, 2, NULL)

```

```

insert into Proiecte values (1, '6 FEB 2020', '5 MAR 2020', 'A1', 1)
insert into Proiecte values (2, '8 JAN 2017', '15 OCT 2018', 'B2', 2)
insert into Proiecte values (3, '17 APR 2018', '5 MAR 2020', 'C3', 3)
insert into Proiecte values (4, '21 SEP 2019', '25 DEC 2019', 'D4', 4)
insert into Proiecte values (5, '6 NOV 2019', '19 AUG 2020', 'C2', 2)

```

```

insert into Produse values (1, 'API', 4325, 1)
insert into Produse values (2, 'Horizon Zero Dawn', 325, 2)
insert into Produse values (3, 'Horizon West', 400, 2)
insert into Produse values (4, 'RE Software', 3600, 3)
insert into Produse values (5, 'ACD Home', 340, 4)

```

```

insert into Comenzi values (1, 1, 2, 4)
insert into Comenzi values (6, 2, 2, 2)
insert into Comenzi values (10, 3, 3, 5)
insert into Comenzi values (2, 4, 4, 5)
insert into Comenzi values (1, 5, 5, 2)
insert into Comenzi values (1, 6, 1, 3)
insert into Comenzi values (1, 7, 2, 3)
insert into Comenzi values (1, 8, 3, 4)

```