## Calbeaza Mihaela Maria

MicroProiect Baze de Date

1. Schema:

Diagram

Description automatically generated

2. Codul sursa:

CREATE DATABASE COMPANIE

USE COMPANIE

---TABELA ANGAJATIILOR CARE CUPRINDE id\_angajat- cheie primara + alte date despre angajat

---id\_sef - id-ul sefului de departament

---id\_departament - id-ul departamentului la care lucreaza fiecare angajat

CREATE TABLE angajat (

id\_angajat INT PRIMARY KEY,

nume VARCHAR(40),

prenume VARCHAR(40),

data\_nasterii DATE,

sex VARCHAR(10),

salar INT,

id\_sef INT,

id\_departament INT

);

---TABELA DEPARTAMENTELOR CUPRINDE id\_departament- cheie primara (face referire la tabela angajat)

--- id\_manager- id-ul managerului de departament si data la care acesta si-a inceput munca

---id\_manager este cheie secundara care face referire la id\_angajat din tabela angajat

CREATE TABLE departament (

id\_departament INT PRIMARY KEY,

nume\_departament VARCHAR(40),

id\_manager INT,

manager\_data\_inceput DATE,

FOREIGN KEY(id\_manager) REFERENCES angajat(id\_angajat) ON DELETE SET NULL

);

---an facut alte legaturi intre tabele

ALTER TABLE angajat

ADD FOREIGN KEY(id\_departament)

REFERENCES departament(id\_departament) ---legatura dintre angajat din tabela angajat si departamentul la care lucreaza din tabela departament

ON DELETE SET NULL;

---TABELA CLIENTIILOR CUPRINDE id\_client- cheie primara(id-ul fiecarui client)

---id\_departament care face referire la id\_departament din tabela departament

CREATE TABLE client (

id\_client INT PRIMARY KEY,

nume\_client VARCHAR(40),

id\_departament INT,

FOREIGN KEY(id\_departament) REFERENCES departament(id\_departament) ON DELETE SET NULL

);

---TABELA ECHIPELOR CUPRINDE id\_angajat si id\_client sunt campuri care formeaza cheia primara

---am facut referire la id\_angajat si id\_client din tabela angajat, respectiv client

CREATE TABLE echipe (

id\_angajat INT,

id\_client INT,

total\_vanzari INT,

PRIMARY KEY(id\_angajat, id\_client),

FOREIGN KEY(id\_angajat) REFERENCES angajat(id\_angajat) ON DELETE CASCADE,

FOREIGN KEY(id\_client) REFERENCES client(id\_client) ON DELETE CASCADE

);

---TABELA FURNIZORIILOR CUPRINDE id\_departament si numa\_furnizor campuri care formeaza cheia primara

---am facut referire la id\_departament din tebla departament

CREATE TABLE furnizori (

id\_departament INT,

nume\_furnizor VARCHAR(40),

tip\_produs VARCHAR(40),

PRIMARY KEY(id\_departament, nume\_furnizor),

FOREIGN KEY(id\_departament) REFERENCES departament(id\_departament) ON DELETE CASCADE

);

-- -----------------------------------------------------------------------------

-- Popularea tabelelor

INSERT INTO angajat VALUES(100, 'Ion', 'Creanga', '1967-11-17', 'M', 250000, NULL, NULL);

INSERT INTO departament VALUES(1, 'PR', 100, '2006-02-09');

UPDATE angajat

SET id\_departament = 1

WHERE id\_angajat = 100;

INSERT INTO angajat VALUES(101, 'Otilia', 'Marculescu', '1961-05-11', 'F', 110000, 100, 1);

-- Marketing

INSERT INTO angajat VALUES(102, 'George', 'Calinescu', '1964-03-15', 'M', 75000, 100, NULL);

INSERT INTO departament VALUES(2, 'Marketing', 102, '1992-04-06');

UPDATE angajat

SET id\_departament = 2

WHERE id\_angajat = 102;

INSERT INTO angajat VALUES(103, 'Ileana', 'Cosanzeana', '1971-06-25', 'F', 63000, 102, 2);

INSERT INTO angajat VALUES(104, 'Zhavia', 'Ward', '1980-02-05', 'F', 55000, 102, 2);

INSERT INTO angajat VALUES(105, 'Fat', 'Frumos', '1958-02-19', 'M', 69000, 102, 2);

-- Financiar

INSERT INTO angajat VALUES(106, 'Costache', 'Giurguveanu', '1969-09-05', 'M', 78000, 100, NULL);

INSERT INTO departament VALUES(3, 'Gheorgita Andrei', 106, '1998-02-13');

UPDATE angajat

SET id\_departament = 3

WHERE id\_angajat = 106;

INSERT INTO angajat VALUES(107, 'Andi', 'Moisescu', '1973-07-22', 'M', 65000, 106, 3);

INSERT INTO angajat VALUES(108, 'Niall', 'Horan', '1978-10-01', 'M', 71000, 106, 3);

-- Furnizorii departamentelor

INSERT INTO furnizori VALUES(2, 'DACO', 'Hartie');

INSERT INTO furnizori VALUES(2, 'Parker', 'Ustensile de Scris');

INSERT INTO furnizori VALUES(3, 'Premier', 'Harnie');

INSERT INTO furnizori VALUES(2, 'RUVIX', 'Formulare Personalizate');

INSERT INTO furnizori VALUES(3, 'Parker', 'Ustensile de Scris');

INSERT INTO furnizori VALUES(3, 'Canon', 'Hartie');

INSERT INTO furnizori VALUES(3, 'Campia Express', 'Formulare personalizate');

-- CLIENT

INSERT INTO client VALUES(400, 'UTCN', 2);

INSERT INTO client VALUES(401, 'Babes-Bolyai', 2);

INSERT INTO client VALUES(402, 'Sameday', 3);

INSERT INTO client VALUES(403, 'Tuca Zbarcea si ascociatii', 3);

INSERT INTO client VALUES(404, 'Turda News', 2);

INSERT INTO client VALUES(405, 'Ziarul Adevarul', 3);

INSERT INTO client VALUES(406, 'Sameday', 2);

-- echipele:

INSERT INTO echipe VALUES(105, 400, 55000);

INSERT INTO echipe VALUES(102, 401, 267000);

INSERT INTO echipe VALUES(108, 402, 22500);

INSERT INTO echipe VALUES(107, 403, 5000);

INSERT INTO echipe VALUES(108, 403, 12000);

INSERT INTO echipe VALUES(105, 404, 33000);

INSERT INTO echipe VALUES(107, 405, 26000);

INSERT INTO echipe VALUES(102, 406, 15000);

INSERT INTO echipe VALUES(105, 406, 130000);

----------------------------------------------------------------------------------------------------------------------------

-- Toti angajatii

SELECT \*

FROM angajat;

-- Toti clientii

SELECT \*

FROM client;

-- Toti angajatii ordonati dupa salar

SELECT \*

from angajat

ORDER BY salar DESC;

-- toti angajatoo ordonati dupa sex, in functie de numele lor

SELECT \*

from angajat

ORDER BY sex, nume;

-- numele si prenumele angajatiilor

SELECT nume, angajat.prenume

FROM angajat;

--gaseste toti angajatii care sunt in departamentul 3

SELECT \*

FROM angajat

WHERE id\_departament = 3;

--gaseste toti angajatii care sunt femei si sunt nascute dupa 1969 sau castiga peste 80000lei

SELECT \*

FROM angajat

WHERE (data\_nasterii >= '1970-01-01' AND sex = 'F') OR salar > 80000;

--gaseste toti angajatii nascuti intre anii 1970 si 1975

SELECT \*

FROM angajat

WHERE data\_nasterii BETWEEN '1970-01-01' AND '1975-01-01';

-- media salariilor angajatiilor

SELECT AVG(salar) as media\_salar

FROM angajat;

-- suma salarului tuturor angajatiilor

SELECT SUM(salar) as total\_salar

FROM angajat;

-- cati angajati de fiecare sex sunt

SELECT COUNT(sex) as nu\_de\_fiecare\_sex, sex

FROM angajat

GROUP BY sex

---suma totala de bani cheltuita de fiecrae client

SELECT SUM(total\_vanzari), id\_client

FROM echipe

GROUP BY id\_client;

-- gaseste angajatul care e nascut in luna a 9a a anului

SELECT \*

FROM angajat

WHERE data\_nasterii LIKE '\_\_\_\_\_09%';

----------------------------------------------------------------------------------------------------------------------------

---afiseaza toti clientii si furnizorii fiecarui departament

SELECT client.nume\_client AS Clienti, client.id\_departament AS Departament\_ID

FROM client

UNION

SELECT furnizori.nume\_furnizor, furnizori.id\_departament

FROM furnizori;

---am adaugat o noua inregistrare in tabelul departament

INSERT INTO departament VALUES (4, 'Secretariat', NULL, NULL);

select\*from departament

---afiseaza numele sefului fiecarui departament

SELECT angajat.id\_angajat, angajat.nume, angajat.prenume, departament.nume\_departament

FROM angajat

JOIN departament

ON angajat.id\_angajat = departament.id\_manager;

-- numele tuturor angajatiilor care au vandut peste 50,000

SELECT angajat.nume, angajat.prenume

FROM angajat

WHERE angajat.id\_angajat IN (SELECT echipe.id\_angajat

FROM echipe

WHERE echipe.total\_vanzari > 50000);

---gaseste toti clietii departamentului condus de George Calinescu

---stiu id-ul lui George Calinescu

SELECT client.id\_client, client.nume\_client

FROM client

WHERE client.id\_departament = (SELECT departament.id\_departament

FROM departament

WHERE departament.id\_manager = 102);

---gaseste toti clietii departamentului condus de George Calinescu

---nu stiu id-ul lui George Calinescu

SELECT client.id\_client, client.nume\_client

FROM client

WHERE client.id\_departament = (SELECT departament.id\_departament

FROM departament

WHERE departament.id\_manager = (SELECT TOP 3 angajat.id\_angajat

FROM angajat

WHERE angajat.nume = 'George' AND angajat.prenume ='Calinescu'

));

--gaseste numele angajatiilor care lucreaza cu clientii departamentului financiar

SELECT angajat.nume, angajat.prenume

FROM angajat

WHERE angajat.id\_angajat IN (

SELECT echipe.id\_angajat

FROM echipe

)

AND angajat.id\_departament = 3;

-- numele clientiilor care au chelatuit mau mult de 100,000 RON

SELECT client.nume\_client

FROM client

WHERE client.id\_client IN (

SELECT id\_client

FROM (

SELECT SUM(echipe.total\_vanzari) AS total, id\_client

FROM echipe

GROUP BY id\_client) AS vanzari\_totale\_client

WHERE total > 100000

);

---------------------------------------------

--------PROCEDURE

CREATE PROCEDURE procedura2 @id\_angajat int, @nume varchar(40), @prenume varchar(40), @nastere date, @sex varchar(10),@salar int, @id\_departament int as

if exists (select\* from angajat where angajat.id\_angajat=@id\_angajat)

begin

insert angajat(id\_angajat, nume, prenume, data\_nasterii, sex, salar, id\_departament)values(@id\_angajat, @nume, @prenume, @nastere, @sex, @salar, @id\_departament)

select\*from angajat

end

else

begin

insert angajat(id\_angajat, nume, prenume, data\_nasterii, sex, salar, id\_departament)values(@id\_angajat, @nume, @prenume, @nastere, @sex, @salar, @id\_departament)

insert departament(id\_departament) values(@id\_departament)

select\*from angajat

end

go

EXEC procedura2 '198', 'Gomez', 'Selena', '2000-06-09', 'undefined', 1000, 4

------------------------------------------------------------------------------------------------------------------------------------

----TRIGGERE-----------------

--UN TABEL IN CARE SE AFISEAZA DATELE DIN URMA EXECUTARII TRIGGERELOR

CREATE TABLE testTrigger (

message VARCHAR(100)

);

---IN MOMENTUL IN CARE INSEREZ UN NOU ANGAJAT O SA-MI AFISEZE UN MESAJ IN TEBALELUL testTrigger

---si o sa-mi afiseze si prenumele angajatului nou introdus

CREATE TRIGGER trigger2 ON angajat AFTER INSERT

BEGIN

FOR EACH ROW

INSERT INTO testTrigger VALUES(NEW.prenume);

end

INSERT INTO angajat VALUES(115, 'Post', 'Malone', '1978-02-19', 'M', 69000, 106, 3);

SELECT\*FROM testTrigger;

---IN MOMENTUL IN CARE INSEREZ UN NOU ANGAJAT O SA-MI AFISEZE UN MESAJ IN TEBALELUL testTrigger IN FUNCTIE DE SEXUL ANGAJATULUI

CREATE TRIGGER trigger3 ON COMPANIE.angajati AFTER INSERT

FOR EACH ROW BEGIN

IF NEW.sex = 'M' BEGIN

INSERT INTO testTrigger VALUES('s-a adaugat un angajat de sex masculin');

END

ELSE

IF NEW.sex = 'F' BEGIN

INSERT INTO testTrigger VALUES('s-a adaugat un angajat de sex feminin');

END

ELSE

BEGIN

INSERT INTO testTrigger VALUES('s-au adaugat angajati cu sex nespecificat');

END

END

INSERT INTO angajat VALUES(111, 'Pam', 'Beesly', '1988-02-19', 'F', 69000, 106, 3);

SELECT\*FROM testTrigger;

-------------------------------------------------------------------------------------------------------------------------------------

---Functie

CREATE FUNCTION functie1 (@furnizor\_departament varchar(40))

RETURNS INT

AS

BEGIN

DECLARE @id\_furnizordep INT

SET @id\_furnizordep = (SELECT t1.id\_departament FROM furnizori t1 WHERE t1.nume\_furnizor=@furnizor\_departament)

RETURN (SELECT count (t2.id\_departament) AS nu\_servicii\_oferite FROM departament t2 WHERE t2.id\_departament = @id\_furnizordep)

END

SELECT [dbo].functie1('Parker') AS nr\_de\_departamente

-------------------------------------------------------------------------------------------------------------------------------------

---CURSOR---------

DECLARE ziiDeNastere CURSOR

FOR SELECT

nume,

prenume,

data\_nasterii

FROM angajat

DECLARE @nume\_angajat varchar(40), @prenume\_angajat varchar(40), @data\_nasterii date

OPEN ziiDeNastere

FETCH NEXT FROM ziiDeNastere INTO @nume\_angajat, @prenume\_angajat, @data\_nasterii

WHILE @@FETCH\_STATUS = 0

BEGIN

print 'Angajatul ' + @nume\_angajat + ' '+ @prenume\_angajat + ' isi sarbatoreste data nasterii in: '

print @data\_nasterii

FETCH NEXT FROM ziiDeNastere INTO @nume\_angajat, @prenume\_angajat, @data\_nasterii

END