BAZA DE DATE A UNEI COMUNE

Florete Fabian-Andrei

grupa 234

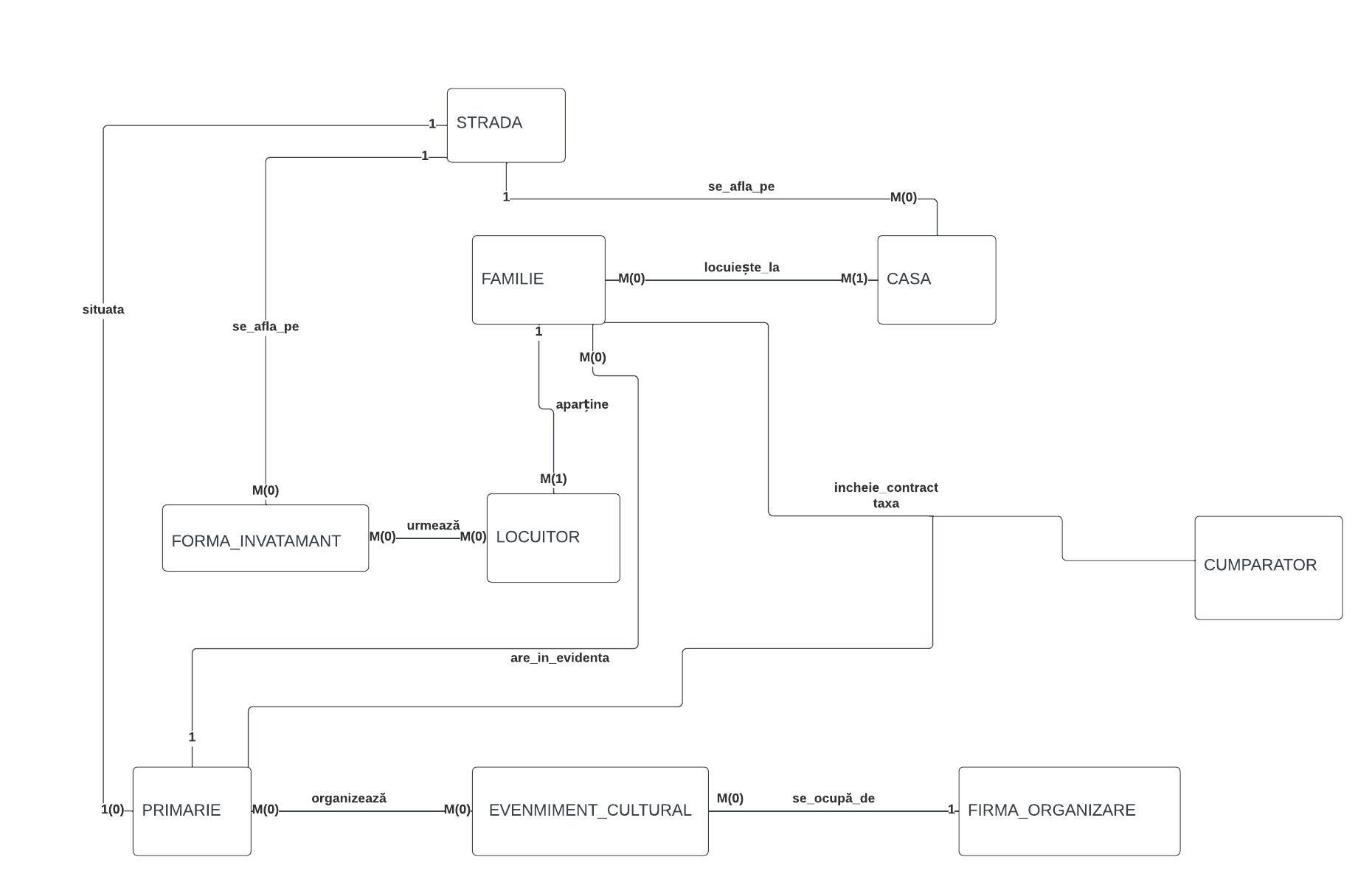
Facultatea de matematică și informatică,

Universitatea din București

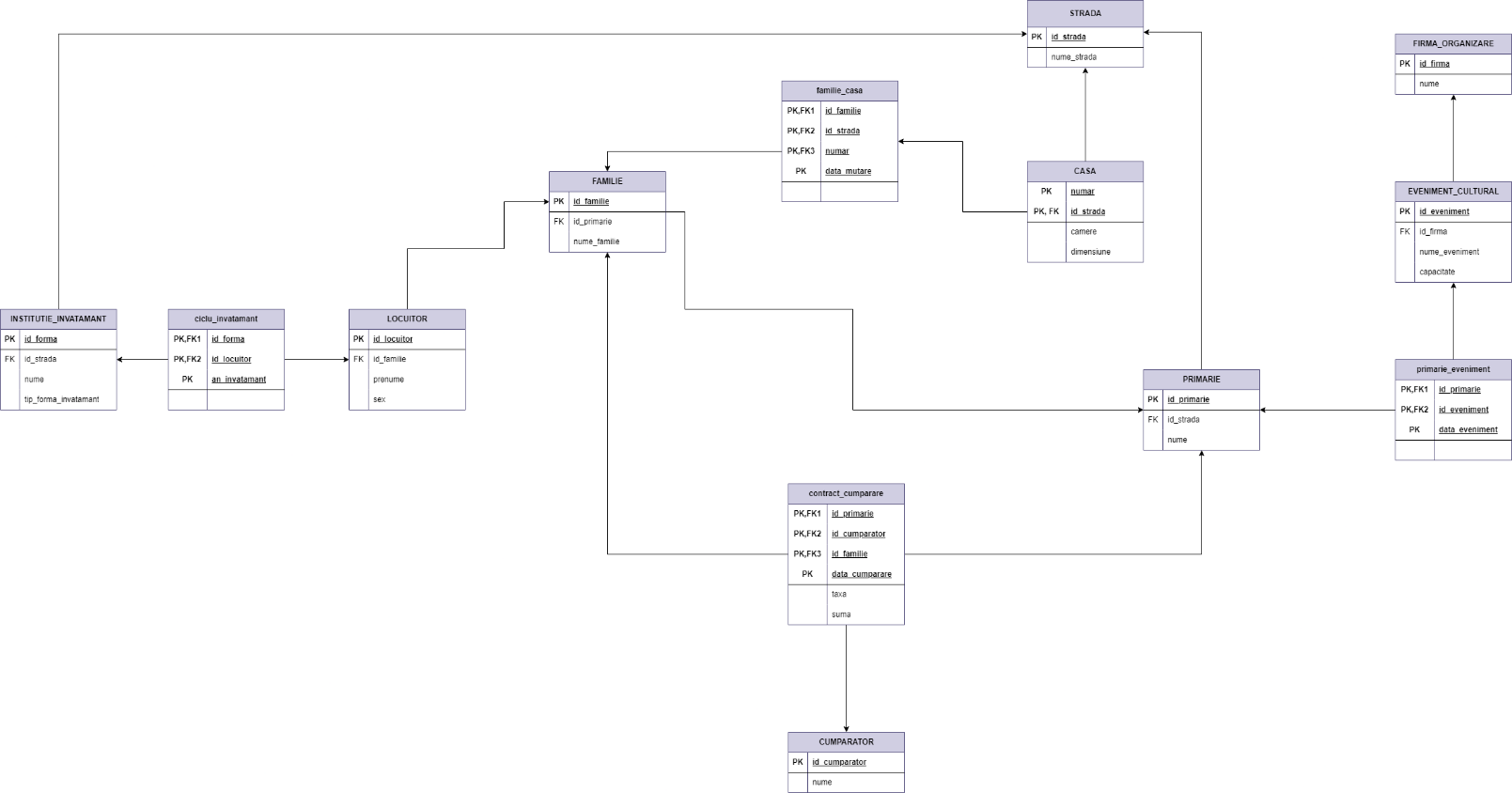
Cerința 1:

Modelul prezentat își propune să modeleze o bază de date corespunzătoare unei comune. Ea are în evidență diferitele familii și cetățeni ce locuiesc în acesta. Primăria satului poate organiza diferite evenimente culturale, precum și oficializa contractele de negoț dintre familii și cumpărători. Pe parcursul timpului, se rețin date precum formele de învățământ absolvite de persoane, precum și casele în care au stat.

Modelul real are ca scop crearea unei baze de date capabilă să gestioneze evenimentele ce se pot petrece într-o comună. El are utilitate în lumea reală, deoarece poate unor entități organizatorice, precum primăriile să-și gestioneze mai ușor cetățenii, urmărind cu ușurință informații precum sumele pe care aceștia le încasează, dar și casele la care locuiesc aceștia.

Cerința 2:

Cerința 3:



Cerința 4:

CREATE TABLE STRADA(

id\_strada NUMBER(3) PRIMARY KEY,

nume\_strada VARCHAR(100) NOT NULL

);

CREATE TABLE INSTITUTIE\_INVATAMANT (

id\_forma NUMBER(3) PRIMARY KEY,

nume VARCHAR(50) NOT NULL,

tip\_forma\_invatamant CHAR(40) NOT NULL,

id\_strada REFERENCES STRADA

);

CREATE TABLE CASA (

id\_strada REFERENCES STRADA,

numar NUMBER(3) NOT NULL CHECK (numar > 0 AND numar < 50),

camere NUMBER(2) DEFAULT 1 NOT NULL,

dimensiune NUMBER(4) NOT NULL,

PRIMARY KEY(id\_strada, numar)

);

CREATE TABLE PRIMARIE (

id\_primarie NUMBER(3) PRIMARY KEY,

nume VARCHAR(100) NOT NULL,

id\_strada REFERENCES STRADA,

CONSTRAINT primarie\_locatie\_unic UNIQUE (id\_strada)

);

CREATE TABLE FAMILIE(

id\_familie NUMBER(4) PRIMARY KEY,

id\_primarie REFERENCES PRIMARIE,

nume\_familie VARCHAR(25) NOT NULL

);

CREATE TABLE LOCUITOR (

id\_locuitor NUMBER(5) PRIMARY KEY,

id\_familie NUMBER(4) REFERENCES FAMILIE(id\_familie),

prenume VARCHAR(50) NOT NULL,

sex NUMBER(1) CHECK (sex > 0 AND sex < 3)

);

CREATE TABLE FIRMA\_ORGANIZARE(

id\_firma NUMBER(3) PRIMARY KEY,

nume VARCHAR(100) NOT NULL

);

CREATE TABLE EVENIMENT\_CULTURAL(

id\_eveniment NUMBER(3) PRIMARY KEY,

id\_firma REFERENCES FIRMA\_ORGANIZARE,

nume\_eveniment VARCHAR(25) NOT NULL,

capacitate NUMBER(7) NOT NULL

);

CREATE TABLE CUMPARATOR(

id\_cumparator NUMBER(3) PRIMARY KEY,

nume VARCHAR(100) NOT NULL

);

CREATE TABLE CICLU\_INVATAMANT(

id\_forma REFERENCES INSTITUTIE\_INVATAMANT,

id\_locuitor REFERENCES locuitor,

an\_invatamant DATE NOT NULL,

PRIMARY KEY (id\_forma, id\_locuitor, an\_invatamant)

);

CREATE TABLE CONTRACT\_CUMPARARE(

id\_primarie REFERENCES PRIMARIE,

id\_cumparator REFERENCES CUMPARATOR,

id\_familie REFERENCES FAMILIE,

data\_cumparare DATE NOT NULL,

taxa NUMBER(3) DEFAULT NULL,

suma NUMBER(8) NOT NULL,

PRIMARY KEY (id\_primarie, id\_cumparator, id\_familie, data\_cumparare)

);

CREATE TABLE FAMILIE\_CASA(

id\_familie NUMBER(4) NOT NULL,

id\_strada NUMBER(3),

numar NUMBER(3),

data\_mutare DATE NOT NULL,

PRIMARY KEY (id\_familie, id\_strada, numar, data\_mutare),

FOREIGN KEY (id\_strada, numar) REFERENCES CASA(id\_strada, numar),

FOREIGN KEY (id\_familie) REFERENCES FAMILIE(id\_familie)

);

CREATE TABLE PRIMARIE\_EVENIMENT(

id\_primarie REFERENCES PRIMARIE,

id\_eveniment REFERENCES EVENIMENT\_CULTURAL,

data\_eveniment DATE NOT NULL,

PRIMARY KEY(id\_primarie, id\_eveniment, data\_eveniment)

);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Cerința 5:

CREATE TABLE STRADA(

id\_strada NUMBER(3) PRIMARY KEY,

nume\_strada VARCHAR(100) NOT NULL

);

CREATE TABLE INSTITUTIE\_INVATAMANT (

id\_forma NUMBER(3) PRIMARY KEY,

nume VARCHAR(50) NOT NULL,

tip\_forma\_invatamant CHAR(40) NOT NULL,

id\_strada REFERENCES STRADA

);

CREATE TABLE CASA (

id\_strada REFERENCES STRADA,

numar NUMBER(3) NOT NULL CHECK (numar > 0 AND numar < 50),

camere NUMBER(2) DEFAULT 1 NOT NULL,

dimensiune NUMBER(4) NOT NULL,

PRIMARY KEY(id\_strada, numar)

);

CREATE TABLE PRIMARIE (

id\_primarie NUMBER(3) PRIMARY KEY,

nume VARCHAR(100) NOT NULL,

id\_strada REFERENCES STRADA,

CONSTRAINT primarie\_locatie\_unic UNIQUE (id\_strada)

);

CREATE TABLE FAMILIE(

id\_familie NUMBER(4) PRIMARY KEY,

id\_primarie REFERENCES PRIMARIE,

nume\_familie VARCHAR(25) NOT NULL

);

CREATE TABLE LOCUITOR (

id\_locuitor NUMBER(5) PRIMARY KEY,

id\_familie NUMBER(4) REFERENCES FAMILIE(id\_familie),

prenume VARCHAR(50) NOT NULL,

sex NUMBER(1) CHECK (sex > 0 AND sex < 3)

);

CREATE TABLE FIRMA\_ORGANIZARE(

id\_firma NUMBER(3) PRIMARY KEY,

nume VARCHAR(100) NOT NULL

);

CREATE TABLE EVENIMENT\_CULTURAL(

id\_eveniment NUMBER(3) PRIMARY KEY,

id\_firma REFERENCES FIRMA\_ORGANIZARE,

nume\_eveniment VARCHAR(25) NOT NULL,

capacitate NUMBER(7) NOT NULL

);

CREATE TABLE CUMPARATOR(

id\_cumparator NUMBER(3) PRIMARY KEY,

nume VARCHAR(100) NOT NULL

);

CREATE TABLE CICLU\_INVATAMANT(

id\_forma REFERENCES INSTITUTIE\_INVATAMANT,

id\_locuitor REFERENCES locuitor,

an\_invatamant DATE NOT NULL,

PRIMARY KEY (id\_forma, id\_locuitor, an\_invatamant)

);

CREATE TABLE CONTRACT\_CUMPARARE(

id\_primarie REFERENCES PRIMARIE,

id\_cumparator REFERENCES CUMPARATOR,

id\_familie REFERENCES FAMILIE,

data\_cumparare DATE NOT NULL,

taxa NUMBER(3) DEFAULT NULL,

suma NUMBER(8) NOT NULL,

PRIMARY KEY (id\_primarie, id\_cumparator, id\_familie, data\_cumparare)

);

CREATE TABLE FAMILIE\_CASA(

id\_familie NUMBER(4) NOT NULL,

id\_strada NUMBER(3),

numar NUMBER(3),

data\_mutare DATE NOT NULL,

PRIMARY KEY (id\_familie, id\_strada, numar, data\_mutare),

FOREIGN KEY (id\_strada, numar) REFERENCES CASA(id\_strada, numar),

FOREIGN KEY (id\_familie) REFERENCES FAMILIE(id\_familie)

);

CREATE TABLE PRIMARIE\_EVENIMENT(

id\_primarie REFERENCES PRIMARIE,

id\_eveniment REFERENCES EVENIMENT\_CULTURAL,

data\_eveniment DATE NOT NULL,

PRIMARY KEY(id\_primarie, id\_eveniment, data\_eveniment)

);

DROP SEQUENCE strada\_seq;

CREATE SEQUENCE strada\_seq

START with 1

INCREMENT BY 5

NOCACHE;

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Teilor');

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Libertatii');

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Viilor');

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Crinului');

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Tineretului');

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Eroilor');

INSERT INTO STRADA

VALUES(strada\_seq.nextval, 'Singuratatii');

INSERT INTO STRADA

VALUES(64, 'Metalurgiei');

INSERT INTO CASA

VALUES(1, 1, 5, 420);

INSERT INTO CASA

VALUES(11, 2, 2, 210);

INSERT INTO CASA

VALUES(16, 14, 1, 100);

INSERT INTO CASA

VALUES(21, 4, 3, 250);

INSERT INTO CASA

VALUES(21, 5, 2, 260);

INSERT INTO CASA

VALUES(16, 25, 3, 420);

INSERT INTO CASA

VALUES(64, 13, 2, 140);

INSERT INTO PRIMARIE

VALUES(1, 'Primaria Viilor', 6);

INSERT INTO PRIMARIE

VALUES(2, 'Primaria Tineretului', 16);

INSERT INTO PRIMARIE

VALUES(3, 'Primaria Crinului', 11);

INSERT INTO PRIMARIE

VALUES(4, 'Primaria Eroilor', 21);

INSERT INTO PRIMARIE

VALUES(5, 'Primaria Teilor', 1);

INSERT INTO PRIMARIE

VALUES(6, 'Primaria Degeaba', 26);

INSERT INTO FAMILIE

VALUES(5, 4, 'Popescu');

INSERT INTO FAMILIE

VALUES(10, 1, 'Ionescu');

INSERT INTO FAMILIE

VALUES(15, 1, ' Popa');

INSERT INTO FAMILIE

VALUES(20, 4, 'Constantinescu');

INSERT INTO FAMILIE

VALUES(25, 5, 'Tupeiescu');

INSERT INTO FAMILIE

VALUES(30, 2, 'Vecinescu');

INSERT INTO LOCUITOR

VALUES (1, 5, 'Ion', 1);

INSERT INTO LOCUITOR

VALUES (2, 5,'Maria', 2);

INSERT INTO LOCUITOR

VALUES (3, 5, 'Ana', 2);

INSERT INTO LOCUITOR

VALUES (4, 20, 'Mihai', 1);

INSERT INTO LOCUITOR

VALUES (5, 20, 'Elena', 2);

INSERT INTO LOCUITOR

VALUES(6, 25, 'Costache', 1);

INSERT INTO LOCUITOR

VALUES(7, 25, 'Mihaela', 2);

INSERT INTO LOCUITOR

VALUES(8, 30, 'Ramon', 1);

INSERT INTO INSTITUTIE\_INVATAMANT

VALUES (1, 'Scoala Gimnaziala Nr. 1', 'Scoala Gimnaziala', 1);

INSERT INTO INSTITUTIE\_INVATAMANT

VALUES (2, 'Liceul Teoretic "Mihai Eminescu"', 'Liceu', 1);

INSERT INTO INSTITUTIE\_INVATAMANT

VALUES (3, 'Scoala Gimnaziala Nr. 2', 'Scoala Gimnaziala', 6);

INSERT INTO INSTITUTIE\_INVATAMANT

VALUES (4, 'Colegiul National "Gheorghe Lazar"', 'Colegiu', 16);

INSERT INTO INSTITUTIE\_INVATAMANT

VALUES (5, 'Liceul Tehnic', 'Liceu', 21);

INSERT INTO CUMPARATOR

VALUES (1, 'ABC Company');

INSERT INTO CUMPARATOR

VALUES (2, 'XYZ Corporation');

INSERT INTO CUMPARATOR

VALUES (3, '123 Industries');

INSERT INTO CUMPARATOR

VALUES (4, 'Acme Enterprises');

INSERT INTO CUMPARATOR

VALUES (5, 'Global Solutions');

INSERT INTO FIRMA\_ORGANIZARE

VALUES (1, 'Event Management Company');

INSERT INTO FIRMA\_ORGANIZARE

VALUES (2, 'Art Agency');

INSERT INTO FIRMA\_ORGANIZARE

VALUES (3, 'Production House');

INSERT INTO FIRMA\_ORGANIZARE

VALUES (4, 'Entertainment Solutions');

INSERT INTO FIRMA\_ORGANIZARE

VALUES (5, 'Creative Events');

INSERT INTO EVENIMENT\_CULTURAL

VALUES (1, 1, 'Concert Live', 1000);

INSERT INTO EVENIMENT\_CULTURAL

VALUES (2, 5, 'Expozitie de arta', 500);

INSERT INTO EVENIMENT\_CULTURAL

VALUES (3, 5, 'Teatru in aer liber', 300);

INSERT INTO EVENIMENT\_CULTURAL

VALUES (4, 1, 'Dans contemporan', 800);

INSERT INTO EVENIMENT\_CULTURAL

VALUES (5, 4, 'Film in aer liber', 200);

INSERT INTO CICLU\_INVATAMANT

VALUES(1, 1, '15-sep-1985');

INSERT INTO CICLU\_INVATAMANT

VALUES(5, 1, '20-sep-1993');

INSERT INTO CICLU\_INVATAMANT

VALUES(3, 2, '15-sep-1985');

INSERT INTO CICLU\_INVATAMANT

VALUES(4, 2, '20-sep-1993');

INSERT INTO CICLU\_INVATAMANT

VALUES(3, 3, '29-aug-2003');

INSERT INTO CICLU\_INVATAMANT

VALUES(4, 3, '30-sep-2011');

INSERT INTO CICLU\_INVATAMANT

VALUES(1, 4, '29-aug-2003');

INSERT INTO CICLU\_INVATAMANT

VALUES(4, 4, '30-sep-2011');

INSERT INTO CICLU\_INVATAMANT

VALUES(3, 5, '11-sep-2002');

INSERT INTO CICLU\_INVATAMANT

VALUES(5, 5, '19-sep-2010');

INSERT INTO CICLU\_INVATAMANT

VALUES(4, 6, '20-may-2003');

INSERT INTO FAMILIE\_CASA

VALUES(5, 16, 14, '10-may-1980');

INSERT INTO FAMILIE\_CASA

VALUES(10, 11, 2, '10-april-1985');

INSERT INTO FAMILIE\_CASA

VALUES(5, 21, 4, '9-april-1985');

INSERT INTO FAMILIE\_CASA

VALUES(20, 21, 5, '29-aug-2003');

INSERT INTO FAMILIE\_CASA

VALUES(15, 21, 4, '15-feb-1995');

INSERT INTO FAMILIE\_CASA

VALUES(15, 1, 1,'20-jan-1997');

INSERT INTO FAMILIE\_CASA

VALUES(10, 1, 1, '14-jan-2001');

INSERT INTO FAMILIE\_CASA

VALUES(15, 11, 2, '10-june-2000');

INSERT INTO FAMILIE\_CASA

VALUES(20, 21, 5, '19-jul-2005');

INSERT INTO FAMILIE\_CASA

VALUES(20, 1, 1, '20-march-2002');

INSERT INTO FAMILIE\_CASA

VALUES(10, 21, 4, '15-march-2002');

INSERT INTO FAMILIE\_CASA

VALUES(25, 16, 14, '5-june-2001');

INSERT INTO FAMILIE\_CASA

VALUES(30, 16, 25, '3-may-2001');

INSERT INTO CONTRACT\_CUMPARARE

VALUES(1, 1, 5, '15-sep-1990', 15, 120000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(1, 2, 5, '20-jan-1985', 20, 17000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(3, 1, 20, '15-aug-2019', 5, 7000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(2, 5, 25,'14-sep-2023', 40, 800000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(3, 4, 20, '19-jul-2021', 14, 75189);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(2, 2, 25, '11-june-2021', 10, 10000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(1, 1, 5, '15-sep-2004', 5, 6510);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(3, 2, 20, '14-sep-2022', 10, 19000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(2, 1, 25, '19-oct-2022', 4, 7000);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(1, 3, 15, '18-nov-2022', 10, 19500);

INSERT INTO CONTRACT\_CUMPARARE

VALUES(2, 3, 10, '20-aug-2001', NULL, 10000);

CREATE SEQUENCE id\_prim

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 5

CYCLE

NOCACHE;

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 1, '12-july-1998');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 5, '6-november-2006');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 3, '20-september-1993');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 2, '3-may-2014');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 1, '18-january-2002');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 3, '9-october-2022');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 5, '28-march-2022');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 4, '14-june-2007');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 2, '27-august-1992');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 1, '01-may-2011');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 2, '05-aug-2022');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(id\_prim.nextval, 3, '29-sep-2022');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(1, 4, '1-may-2001');

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(1, 5, '16-aug-2023');

A screenshot of a computer

Description automatically generated

Cerința 6:

--Sa se mareasca contractele de cumparare cu suma >= 10.000 cu inca 10 . Afisati contractele modificate(dupa)

--din acest nou tablou sa se pastreze intr un vector doar familiile care au avut contracte cu cumparatorul cu id 1. Daca nu exista => se afiseaza un mesaj de eroare

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE p\_cerinta6

IS

TYPE tablou\_indexat IS TABLE OF contract\_cumparare%rowtype INDEX BY PLS\_INTEGER;

t\_contracte tablou\_indexat;

TYPE vector IS VARRAY(20) OF contract\_cumparare%rowtype;

t\_vector vector:=vector();

TYPE tablou\_imbricat IS TABLE OF contract\_cumparare%rowtype;

t\_contracte\_sterse tablou\_imbricat := tablou\_imbricat();

BEGIN

SELECT id\_primarie, id\_cumparator, id\_familie, data\_cumparare, taxa, suma BULK COLLECT INTO t\_contracte

FROM contract\_cumparare;

FOR i in t\_contracte.FIRST..t\_contracte.LAST LOOP

IF(t\_contracte(i).suma >= 10000) THEN

t\_vector.extend;

t\_vector(t\_vector.count) := t\_contracte(i);

END IF;

END LOOP;

FOR i in t\_vector.first..t\_vector.last LOOP

t\_contracte\_sterse.extend;

UPDATE contract\_cumparare

SET suma = suma + 0.01\*suma

WHERE id\_cumparator = t\_vector(i).id\_cumparator

AND id\_familie = t\_vector(i).id\_familie

AND data\_cumparare = t\_vector(i).data\_cumparare

RETURNING id\_primarie, id\_cumparator, id\_familie, data\_cumparare, taxa, suma

INTO t\_contracte\_sterse(i);

END LOOP;

FOR i in t\_contracte\_sterse.first..t\_contracte\_sterse.last LOOP

DBMS\_OUTPUT.PUT\_LINE(t\_contracte\_sterse(i).id\_familie || ' ' || t\_contracte\_sterse(i).suma);

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu au fost gasite contracte');

END p\_cerinta6;

/

SELECT \* FROM contract\_cumparare;

BEGIN

p\_cerinta6;

END;

/

SELECT \* FROM contract\_cumparare;

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Cerința 7:

CREATE OR REPLACE PROCEDURE p\_cerinta7

IS

TYPE cursor\_ref IS REF CURSOR;

CURSOR c\_familie IS SELECT id\_familie, nume\_familie,

CURSOR(SELECT l.id\_locuitor, l.prenume

FROM locuitor l

WHERE l.id\_familie = f.id\_familie)

FROM familie f;

v\_id\_familie familie.id\_familie%type;

v\_nume\_familie familie.nume\_familie%type;

v\_id\_locuitor locuitor.id\_locuitor%type;

v\_prenume locuitor.prenume%type;

v\_cursor cursor\_ref;

v\_cnt NUMBER(5);

BEGIN

OPEN c\_familie;

LOOP

FETCH c\_familie INTO v\_id\_familie, v\_nume\_familie, v\_cursor;

EXIT WHEN c\_familie%NOTFOUND;

v\_cnt := 0;

DBMS\_OUTPUT.PUT\_LINE('Afisez date pentru familia: ' || v\_nume\_familie);

LOOP

FETCH v\_cursor INTO v\_id\_locuitor, v\_prenume;

EXIT WHEN v\_cursor%NOTFOUND;

v\_cnt := v\_cnt + 1;

DBMS\_OUTPUT.PUT\_LINE(v\_prenume);

END LOOP;

IF v\_cnt = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Familia nu are locuitori');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Familia are ' || v\_cnt || ' persoane');

END IF;

END LOOP;

CLOSE c\_familie;

END p\_cerinta7;

/

BEGIN

p\_cerinta7();

END;

/

A screenshot of a computer program

Description automatically generated

A blue and white rectangular object with a blue border

Description automatically generated

Cerința 8:  
--aflati strada casei unei familii cu un anumit nume(eroare daca nu exista id-ul),

CREATE OR REPLACE FUNCTION f\_cerinta8

(v\_nume familie.nume\_familie%type)

RETURN strada.nume\_strada%TYPE

IS

v\_nume\_strada strada.nume\_strada%type;

v\_cnt\_familie NUMBER(2);

v\_id\_strada strada.id\_strada%type;

v\_numar\_casa familie\_casa.numar%type;

v\_data familie\_casa.data\_mutare%type;

v\_data2 familie\_casa.data\_mutare%type;

ex\_casa\_ocupata EXCEPTION; -- prima exceptie, in caz ca sta cineva in casa lor deja

ex\_familie\_negasita EXCEPTION;

PRAGMA EXCEPTION\_INIT(ex\_casa\_ocupata, -20001);

PRAGMA EXCEPTION\_INIT(ex\_familie\_negasita, -20002);

BEGIN

--prima data verific daca exita familia sau mai multe chiar cu acelasi nume

SELECT count(nume\_familie) INTO v\_cnt\_familie

FROM familie

WHERE lower(nume\_familie) = lower(v\_nume);

IF v\_cnt\_familie = 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Nu a fost gasita o familie cu acest nume');

END IF;

WITH aux AS(

SELECT ss.id\_strada, fc.numar, fc.data\_mutare

FROM familie\_casa fc JOIN strada ss ON fc.id\_strada = ss.id\_strada JOIN familie f on f.id\_familie = fc.id\_familie

WHERE lower(f.nume\_familie) = lower(v\_nume)

ORDER BY data\_mutare desc

)

SELECT \* INTO v\_id\_strada, v\_numar\_casa, v\_data

FROM aux

WHERE ROWNUM = 1;

--verific daca in ultima lor casa nu cumva sta deja altcineva(lucru cauzat de eroare la validarea datelor)

--gasesc ultima data la care s-a mutat cineva in acea casa

WITH aux as(

SELECT data\_mutare

FROM familie\_casa

WHERE id\_strada = v\_id\_strada AND numar = v\_numar\_casa

ORDER BY data\_mutare desc)

SELECT \* INTO v\_data2

FROM aux

WHERE ROWNUM = 1;

IF v\_data < v\_data2 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'O familie deja sta la aceasta casa');

END IF;

SELECT nume\_strada INTO v\_nume\_strada

FROM strada

WHERE id\_strada = v\_id\_strada;

return v\_nume\_strada;

EXCEPTION

WHEN EX\_FAMILIE\_NEGASITA THEN

DBMS\_OUTPUT.PUT\_LINE('Nu a fost gasita o familie cu acest nume');

return -1;

WHEN EX\_CASA\_OCUPATA THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare la data casei');

return -1;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu a fost gasita familia/casa');

return -1;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Alta eroare');

return -1;

END f\_cerinta8;

/

BEGIN

DBMS\_output.put\_line(f\_cerinta8('Popescu')); --exemplu care produce eroare

DBMS\_output.put\_line(f\_cerinta8('Constantinescu')); -- exemplu care merge bine

DBMS\_output.put\_line(f\_cerinta8('nu exist')); --exemplu in care nu exista familia

END;

/

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Cerința 9:

--Numele institutiilor de invatamant pe care le au absolvit membrii familiei cu cel mai mare contract la o companie data

CREATE OR REPLACE PROCEDURE procedura\_cerinta\_9

(v\_cumparator VARCHAR)

IS

TYPE info is RECORD

(nume\_institutie institutie\_invatamant.nume%type,

prenume locuitor.prenume%type,

nume familie.nume\_familie%type);

TYPE tablou\_indexat IS TABLE OF info INDEX BY pls\_integer;

t tablou\_indexat;

BEGIN

SELECT ii.nume, l.prenume, f.nume\_familie BULK COLLECT INTO t

FROM CUMPARATOR c JOIN CONTRACT\_CUMPARARE ccu ON c.id\_cumparator = ccu.id\_cumparator

JOIN FAMILIE f ON f.id\_familie = ccu.id\_familie JOIN locuitor l ON l.id\_familie = f.id\_familie

JOIN ciclu\_invatamant ci ON ci.id\_locuitor = l.id\_locuitor JOIN institutie\_invatamant ii ON ii.id\_forma = ci.id\_forma

WHERE lower(c.nume) = lower(v\_cumparator)

AND ccu.suma = (SELECT max(suma) FROM contract\_cumparare ccu2 WHERE ccu2.id\_familie = l.id\_familie AND ccu2.id\_cumparator = c.id\_cumparator);

if(t.count > 0) THEN

FOR i IN t.first..t.last LOOP

DBMS\_OUTPUT.PUT\_LINE(t(i).nume || ' ' || t(i).prenume || ' a studiat la ' || t(i).nume\_institutie);

END LOOP;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Nu a fost gasita vreo institutie de invatamant');

END IF;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare');

END procedura\_cerinta\_9;

/

BEGIN

procedura\_cerinta\_9('Global Solutions');

END;

/

A computer screen shot of a blue screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Cerința 10:

CREATE TABLE istoric\_inserturi

(

utilizator VARCHAR2(30),

data\_actualizare date

);

CREATE OR REPLACE TRIGGER trigger\_cerinta\_10

BEFORE INSERT ON PRIMARIE\_EVENIMENT

BEGIN

INSERT INTO ISTORIC\_INSERTURI

VALUES(SYS.LOGIN\_USER, sysdate);

END;

/

INSERT INTO PRIMARIE\_EVENIMENT

VALUES(1, 1, '12-Jan-2023');

SELECT \* FROM istoric\_inserturi;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Cerința 11:

--Fac un declansator care nu las ca valoarea unui contract sa fie marita cu mai mult de 50% din valoarea acestuia

CREATE OR REPLACE TRIGGER trigger\_cerinta\_11

BEFORE UPDATE ON contract\_cumparare

FOR EACH ROW

DECLARE

v\_exceptie EXCEPTION;

BEGIN

IF(:OLD.suma + :OLD.suma\*0.5) > :NEW.suma THEN

RAISE v\_exceptie;

END IF;

EXCEPTION

WHEN v\_exceptie THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Suma noua a crescut cu mai mult de 50%');

DBMS\_OUTPUT.PUT\_LINE('sal');

END;

/

UPDATE contract\_cumparare

SET suma = suma\*0.5 + 1000

WHERE id\_primarie = 1;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Cerința 12:

--Trigger pe LDD care tine minte istoricul actiunilor

CREATE TABLE istoric\_cerinta\_12

(utilizator VARCHAR2(30),

actiune varchar2(50),

data DATE);

CREATE OR REPLACE TRIGGER trigger\_cerinta\_12

BEFORE CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

IF lower(SYS.SYSEVENT) = lower('create') THEN

INSERT INTO ISTORIC\_CERINTA\_12

VALUES(SYS.LOGIN\_USER, 'Inserted in ' || SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);

ELSIF lower(SYS.SYSEVENT) = lower('alter') THEN

INSERT INTO ISTORIC\_CERINTA\_12

VALUES(SYS.LOGIN\_USER, 'Altered table ' || SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);

ELSIF lower(SYS.SYSEVENT) = lower('drop') THEN

INSERT INTO ISTORIC\_CERINTA\_12

VALUES(SYS.LOGIN\_USER, 'Dropped table ' || SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);

END IF;

END;

/

CREATE TABLE test1(atribut varchar2(30));

CREATE TABLE test2(atribut varchar2(30));

DROP TABLE test1;

DROP TABLE test2;

SELECT \* FROM istoric\_cerinta\_12

A computer screen shot of a blue screen

Description automatically generatedA screenshot of a computer

Description automatically generated