*1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.*

In scopul realizarii proiectului la materia Baze de Date, am construit baza de date a unei scoli de dans pe parcursul unui an.

Baza de date contine informatii cu privire la cursantii scolii de dans, instructorii, receptionistii, paznicii si personalul de curatenie ai acestei scoli. Mai mult, aceasta retine informatii despre fiecare stil de dans predat, ora desfasurarii cursurilor, clasele in care sunt organizati cursantii in functie de varsta, salile dedicate fiecarei clase, cladirile unde se afla aceste sali, adresele cladirilor, dar si informatii despre evenimentele si incasarile lunare.

Am creeat cate un tabel pentru fiecare concept, unde fiecare tabel poate fi identificar printr-o cheie primara, cu exceptia instructorilor, receptionistilor, paznicilor si personalul de curatenie, care sunt identificar prin cheia primara a entitatii “angajat”.

De asemenea, sunt utilizate doua tabele asociative: “efectueaza” si “relatie\_parentala”, indicand relatii de grad superior si relatii many-to-many.

*2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.*

In aceasta baza de date sunt respectate urmatoarele constrangeri:

1. O sala a unei cladiri poate avea o singura adresa, insa pot exista una sau mai multe sali la aceeasi adresa.
2. Un sediu ce apartine scolii de dans poate avea mai multe sali. Fiecare sala apartine de un singur sediu.
3. Un cursant apartine obligatoriu unei clase. O clasa pune la un loc obligatoriu mai multi cursanti.
4. Un cursant poate beneficia de un singur abonament lunar. Un abonament trebuie sa apartina neaparat unui cursant.
5. Unei clase si unui stil de dans le sunt dedicate mai multe ore de curs, insa o ora de curs este dedicata unei singure clase si unui singur stil de dans.
6. Un eveniment poate fi organizat o singura data pe luna. Un cursant primeste un singur numar de puncte in functie de locul castigat la eveniment(battle). Aceste 3 entitati sunt legate printr-o relatie careia ii este atribuit un tabel asociativ care memoreaza aceste punctaje, in scopul premierii la sfarsit de an.
7. O ora de curs are o durata nenula, de 1h 30.
8. Un paznic, un angajat responsabil cu curatenia sau un receptionist pot lucra intr-un singur sediu.
9. Un instructor poate tine mai multe ore de curs, insa o ora poate fi tinuta de un singur instructor.

*3. Descrierea entităților, incluzând precizarea cheii primare.*

CURSANT = beneficiarul cursurilor de dans furnizate de catre scoala de dans

*Cheie primara: id\_cursant*

ANGAJAT= angajat al clubului de dans

*Cheie primara: id\_salariat*

Aceasta cheie reprezinta cheia primara pentru urmatoarele subentitati ale entitatii:

INSTRUCTOR= persoana ce este specializata intr-un anumit stil de dans, ce sustine cursuri in cadrul clubului

PAZNIC= persoana a carei responsabilitate este securizarea unui sediu

ANGAJAT CURATENIE= persoana a carei responsabilitate este ingrijirea unui sediu

RECEPTIONIST= persoana a carei responsabilitate este furnizeze informatii de baza legate de sediu, sa verifice abonamentele cursantilor si sa ii indrume catre sala

ABONAMENT= modalitatea de acces a cursantilor in sala

*Cheie primara: tip*

STIL\_DE\_DANS= domeniu de cunoastere care este predat la o ora de curs

*Cheie primara: id\_stil*

ORA\_DE\_CURS= durata desfasurarii unui curs

*Cheie primara: id\_ora*

EVENIMENT= concurs de dans desfasurat in fiecare luna in cadrul unui sediu

*Cheie primara: id\_eveniment*

CLASA= ansamblu de cursanti ce participa simultan la aceeasi ora de curs

*Cheie primara:id\_clasa*

SALA= locul in care o clasa isi desfasoara activitatea

*Cheie primara: id\_sala*

SEDIU=cladire in care se afla mai multe sali

*Cheie primara: id\_sediu*

ADRESA= locatia in care se afla un sediu

*Cheie primara: id\_adresa*

INCASARI= suma de bani produsa de catre un sediu pe luna

*Cheie primara: id\_incasari*

*4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.*

* CURSANT are ABONAMENT =asociaza un abonament ales de cursant catre acesta

*Cardinalitate minima- 1:1*

*Cardinalitate maxima- m:1*

* CURSANT participa la EVENIMENT= are ca scop sa retina punctajul unui cursant obtinut in cadrul unui eveniment

*Cardinalitate minima- 1:1*

*Cardinalitate maxima-m:m*

* CURSANT participa la ORA\_DE\_CURS=reprezinta ora la care trebuie sa se prezinte cursantul

*Cardinalitate minima-1:1*

*Cardinalitate maxima-m:m*

* CURSANT face parte din CLASA=asociaza un elev cu grupa sa

*Cardinalitate minima: 1:1*

*Cardinalitate maxima: m:1*

* CLASA are SALA=asociaza clasa cu sala in care se desfasoara activitatea acelei clase

*Cardinalitate minima:1:1*

*Cardinalitate maxima: 1:m*

* CLASA ii este dedicata ORA\_DE\_CURS=asociaza clasa si ora de curs la care trebuie sa participe

*Cardinalitate minima: 1:m*

*Cardinalitate maxima: 1:m*

* SALA se afla in SEDIU=asociaza sala cu cladirea in care se afla

*Cardinalitate minima: m:1*

*Cardinalitate maxima: m:1*

* SEDIU se afla la ADRESA= asociaza adresa la care este situat sediul

*Cardinalitate minima: 1:1*

*Cardinalitate maxima: m:1*

* STIL\_DE\_DANS este predat la ORA\_DE\_CURS=asociaza ora la care se desfasoara stilul

*Cardinalitate minima: 1:m*

*Cardinalitate maxima: 1:m*

* ORA\_DE\_CURS sustinuta de INSTRUCTOR=asociaza ora la care instructorul isi va desfasura activitatea

*Cardinalitate minima:m:1*

*Cardinalitate maxima: m:1*

* RECEPTIONIST lucreaza in SEDIU=asociaza sediul unde este angajat receptionistul

*Cardinalitate minima:1:1*

*Cardinalitate maxima:m:1*

* PAZNIC lucreaza in SEDIU= asociaza sediul unde este angajat paznicul

*Cardinalitate minima: 1:1*

*Cardinalitate maxima: m:1*

* ANGAJAT\_CURATENIE lucreaza in SEDIU=asociaza sediul unde este angajata persoana responsabila cu curatenia

*Cardinalitate minima:1:1*

*Cardinalitate maxima:m:1*

* SEDIUL produce INCASARI=asociaza suma de bani produsa pe luna de catre sediu

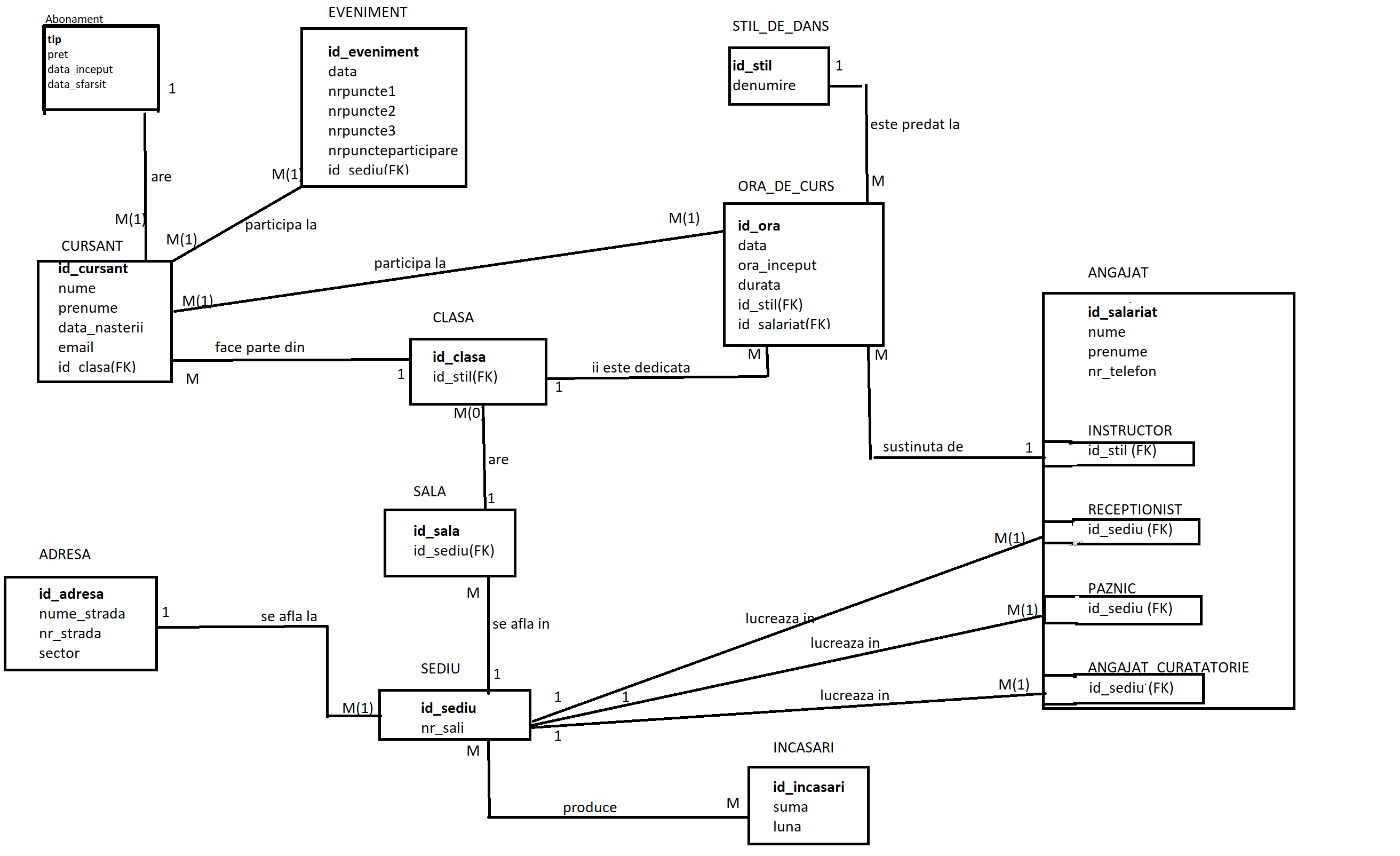
*Cardinalitate minima:1:m*

*Cardinalitate maxima:m:m*

*5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| entitate | atribut | tip de date | constrangeri | valori implicite | valori posibile |
| cursant | id\_cursant | int(10) | not null, unic, |  | 336 |
| nume | varchar(20) | not null |  | Popescu |
| prenume | varchar(20) | not null |  | Ioana |
| data\_nasterii | date | not null |  | 03/06/2003 |
| email | varchar(20) | unic |  | mica@gmail.com |
| angajat | id\_salariat | int(10) | not null, unic |  | 236 |
| nume | varchar(20) | not null |  | Georgescu |
| prenume | varchar(20) | not null |  | Mihai |
| nr\_telefon | varchar(10) | unic |  | 0722346828 |
| email | varchar(20) | unic |  | cineva@gmail.com |
| adresa | id\_adresa | int(10) | not null, unic |  | 12 |
| nume\_strada | varchar(20) | not null |  | Bulevardul Unirii |
| numar\_strada | int(3) | not null |  | 18 |
| sector | int(2) | - |  | 3 |
| clasa | id\_clasa | int(10) | not null, unic |  | 100 |
| sala | id\_sala | int(10) | not null, unic |  | 112 |
| sediu | id\_sediu | char | not null, unic |  | 78 |
| nr\_sali | int(6) | - |  | 10 |
| eveniment | id\_eveniment | int(10) | not null, unic |  | 99 |
| data | date | - |  | 18/06/2022 |
| nrpuncte1 | int(2) |  | 0 | 20 |
| nrpuncte2 | int(2) | - | 0 | 15 |
| nrpuncte3 | int(2) |  | 0 | 10 |
| nrpuncteparticipare | int(2) |  | 0 | 5 |
| stil\_de\_dans | id\_stil | int(10) | not null, unic |  | 54 |
| denumire | varchar(20) | not null |  | dancehall |
| ora\_de\_curs | id\_ora | int(10) | not null, unic |  | 8973 |
| data | date | - | sysdate | 18/05/2022 |
| ora\_inceput | varchar(5) | - | 17:00 | 18:00 |
| durata | int(3) | - | 90 | 90 |
| incasari | id\_incasari | int(10) |  |  | 1111 |
| suma | int(6) |  |  | 10000 |
| luna | int(2) |  |  | 3 |
| abonament | tip | varchar(20) |  |  | nelimitat |
| pret | int(3) |  |  | 200 |
| data\_inceput | date | sysdate |  | 5/04/2022 |
| data\_sfarsit | date | sysdate |  | 5/05/2022 |

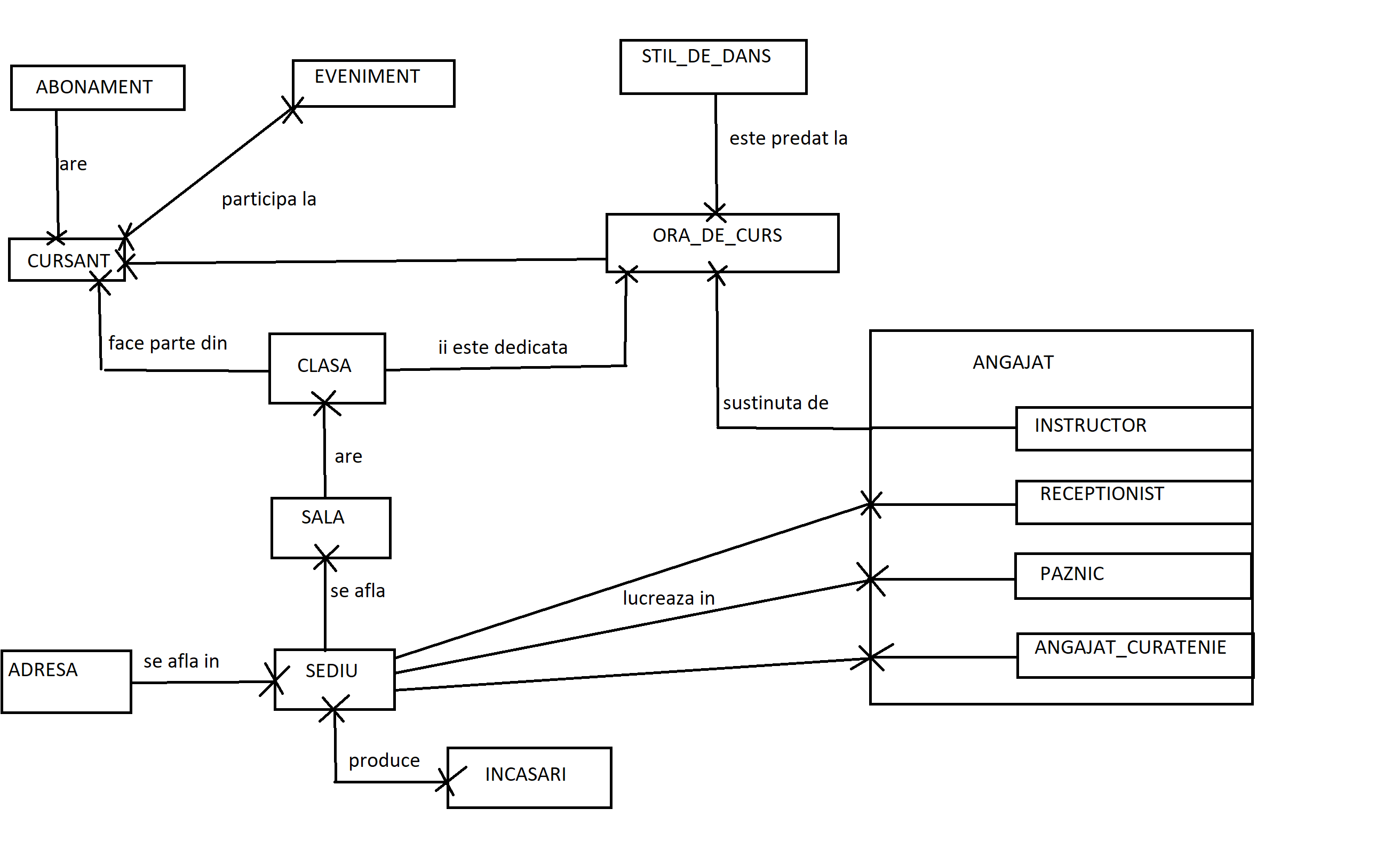
*6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.*

**

*7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la*

*punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără*

*considerarea subentităților), dintre care cel puțin un tabel asociativ.*



*8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la*

*punctul 7*

CURSANT(id\_cursant#, nume, prenume, data\_nasterii, email, id\_clasa#)

ABONAMENT(tip#,pret,data\_inceput,data\_sfarsit)

EVENIMENT(id\_eveniment#,data,nrpuncte1,nrpuncte2,nrpuncte3,nrpuncteparticipare,id\_sediu#)

ORA\_DE\_CURS(id\_ora#,data,ora\_inceput,durata,id\_stil#,id\_salariat#)

STIL\_DE\_DANS(id\_stil#,denumire)

CLASA(id\_clasa#,id\_stil#,id\_sala#)

SALA(id\_sala#,id\_sediu#)

SEDIU(id\_sediu#,nr\_sali)

ADRESA(id\_adresa#,nume\_strada,nr\_strada,sector)

INCASARI(id\_incasari#,suma,luna,id\_sediu#)

ANGAJAT(id\_salariat#,nume,prenume,nr\_telefon)

INSTRUCTOR(id\_stil#)

RECEPTIONIST(id\_sediu#)

PAZNIC(id\_sediu#)

ANGAJAT\_CURATENIE(id\_sediu#)

*9. Realizarea normalizării până la forma normală 3 (FN1-FN3).*

FN1

O relatie este in forma normala 1 daca fiecare atribut care o compune are drept corespondent o valoare indivizibila.

Exemplu FN1 pe relatia ABONAMENT(1)-CURSANT(M(1)):

|  |  |
| --- | --- |
| ABONAMENT(abonament.tip) | CURSANT(cursant.nume, cursant.prenume) |
| nelimitat | Popescu Alex |
| 8 sedinte | Georgescu Ioana |
| nelimitat | Mihai Oana |
| o sedinta | Ionescu Ion |
| 8 sedinte | Dobre Valentina |
| o sedinta | Ilie Maria |
| nelimitat | Trandafir Patricia |

Exemplu non-FN1 pe relatia ABONAMENT(1)-CURSANT(M(1)):

|  |  |
| --- | --- |
| ADRESA(adresa.id\_adresa) | ELEV(elev.nume, elev.prenume) |
| nelimitat | Popescu Alex, Mihai Oana, Trandafir Patricia |
| o sedinta | Ionescu Ion, Ilie Maria |
| 8 sedinte | Georgescu Ioana, Dobre Valentina |

FN2

O relatie este in forma normala 2 daca se afla in FN1 si fiecare atribut care nu este cheie primara este dependent de intreaga cheie primara.

Exemplu FN2 pe tabelul asociativ ORA\_DE\_CURS + tabelul STIL\_DE\_DANS:

|  |  |
| --- | --- |
| id\_ora# | id\_stil# |
| 1 | 5 |
| 2 | 4 |
| 3 | 2 |

|  |  |
| --- | --- |
| id\_stil# | denumire |
| 5 | Informatica |
| 4 | Engleza |
| 2 | Matematica |

Exemplu non-FN2 pe tabelul ORA\_DE\_CURS:

|  |  |  |
| --- | --- | --- |
| id\_ora# | id\_stil# | denumire (stil\_de\_dans) |
| 1 | 5 | Dancehall |
| 2 | 4 | Hip-Hop |
| 3 | 2 | Locking |

FN3

O relatie este in forma normala 3 daca si numai daca este in FN2 si fiecare atribut care nu este cheie depinde direct de cheia primara.

Exemplu FN3 pe tabelul CURSANT+ tabelul ABONAMENT:

|  |  |  |
| --- | --- | --- |
| tip # | data\_inceput | data\_sfarsit |
| nelimitat | 03-JUNE-2022 | 03-JULY-2022 |
| 8 sedinte | 18-JUNE-2022 | 18-JULY-2022 |
| 4 sedinte | 10-JUNE-2022 | 10-JULY-2022 |

Exemplu non-FN3 pe tabelul CURSANT:

|  |  |  |
| --- | --- | --- |
| id\_cursant # | data\_inceput | data\_sfarsit |
| 1 | 03-JUNE-2022 | 03-JULY-2022 |
| 2 | 18-JUNE-2022 | 18-JULY-2022 |
| 3 | 10-JUNE-2022 | 10-JULY-2022 |

1. *Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).*

Pentru a atribui corect id-uri unice pentru un tabel, am construit o secventa „GENERATOR” ce ne va ajuta sa generam cate un id nou pentru fiecare valoare introdusa in tabel.

CREATE SEQUENCE GENERATOR START WITH 1

INCREMENT BY 1

MAXVALUE 100000

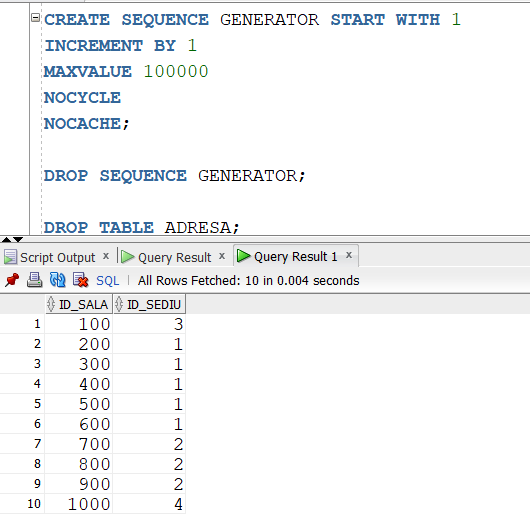
NOCYCLE

NOCACHE;

Pentru a avea id-uri diferite in fiecare tabel, am folosit sintaxa

DROP SEQUENCE GENERATOR

ce are ca scop stergerea generatorului nostru de id-uri, apoi am facut din nou cererea pentru crearea seventei, cu alt numar de inceput si alt pas.



*10. Crearea tabelelor în SQLși inserarea de date coerente în fiecare dintre acestea (minimum5înregistrări înfiecare tabelneasociativ; minimum10înregistrări întabeleleasociative).*

Dupa ce avem generatorul, incepem sa cream efectiv tabelele si sa adaugam inregistrari in ele:

CREATE TABLE ABONAMENT

(TIP VARCHAR(20) PRIMARY KEY, PRET NUMBER(10), DATA\_INCEPUT DATE, DATA\_SFARSIT DATE);

INSERT INTO ABONAMENT VALUES

('NELIMITAT',200,SYSDATE,SYSDATE);

INSERT INTO ABONAMENT VALUES

('12 SEDINTE',180,SYSDATE,SYSDATE);

INSERT INTO ABONAMENT VALUES

('8 SEDINTE',150,SYSDATE,SYSDATE);

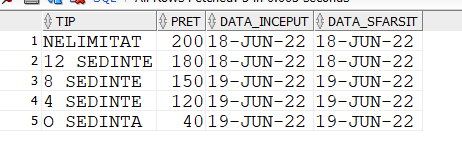
INSERT INTO ABONAMENT VALUES

('4 SEDINTE',120,SYSDATE,SYSDATE);

INSERT INTO ABONAMENT VALUES

('O SEDINTA',40,SYSDATE,SYSDATE);

SELECT \* FROM ABONAMENT;



CREATE TABLE ADRESA (ID\_ADRESA NUMBER(10) PRIMARY KEY,

NUME\_STRADA VARCHAR(20), NR\_STRADA NUMBER(10), SECTOR NUMBER(10));

INSERT INTO ADRESA VALUES

(GENERATOR.NEXTVAL,'BD. UNIRII',11,3);

INSERT INTO ADRESA VALUES

(GENERATOR.NEXTVAL,'ALEEA TANDAREI',18,3);

INSERT INTO ADRESA VALUES

(GENERATOR.NEXTVAL,'STRADA ISTRIEI',20,3);

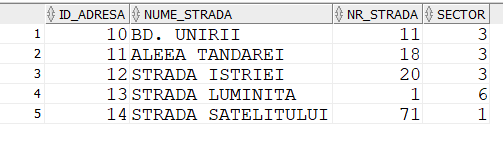
INSERT INTO ADRESA VALUES

(GENERATOR.NEXTVAL,'STRADA LUMINITA',1,6);

INSERT INTO ADRESA VALUES

(GENERATOR.NEXTVAL,'STRADA SATELITULUI',71,1);

SELECT \* FROM ADRESA;



CREATE TABLE SEDIU

(ID\_SEDIU NUMBER(10) PRIMARY KEY, NR\_SALI NUMBER(10) NOT NULL);

INSERT INTO SEDIU VALUES

(GENERATOR.NEXTVAL,10);

INSERT INTO SEDIU VALUES

(GENERATOR.NEXTVAL,5);

INSERT INTO SEDIU VALUES

(GENERATOR.NEXTVAL,1);

INSERT INTO SEDIU VALUES

(GENERATOR.NEXTVAL,3);

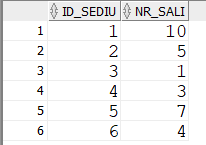
INSERT INTO SEDIU VALUES

(GENERATOR.NEXTVAL,7);

INSERT INTO SEDIU VALUES

(GENERATOR.NEXTVAL,4);

SELECT \* FROM SEDIU;



CREATE TABLE STIL\_DE\_DANS

(ID\_STIL NUMBER(10) PRIMARY KEY, DENUMIRE VARCHAR(20));

INSERT INTO STIL\_DE\_DANS VALUES

(GENERATOR.NEXTVAL, 'DANCEHALL');

INSERT INTO STIL\_DE\_DANS VALUES

(GENERATOR.NEXTVAL, 'HIP HOP');

INSERT INTO STIL\_DE\_DANS VALUES

(GENERATOR.NEXTVAL, 'LOCKING');

INSERT INTO STIL\_DE\_DANS VALUES

(GENERATOR.NEXTVAL, 'HOUSE');

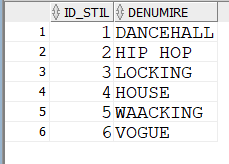
INSERT INTO STIL\_DE\_DANS VALUES

(GENERATOR.NEXTVAL, 'WAACKING');

INSERT INTO STIL\_DE\_DANS VALUES

(GENERATOR.NEXTVAL, 'VOGUE');

SELECT \* FROM STIL\_DE\_DANS;



CREATE TABLE SALA

(ID\_SALA NUMBER(10) PRIMARY KEY, ID\_SEDIU NUMBER(10) REFERENCES SEDIU(ID\_SEDIU));

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 3);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 1);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 1);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 1);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 1);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 1);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 2);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 2);

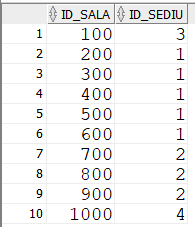
INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 2);

INSERT INTO SALA VALUES

(GENERATOR.NEXTVAL, 4);

SELECT \* FROM SALA;



CREATE TABLE CLASA (ID\_CLASA NUMBER(10) PRIMARY KEY,

ID\_STIL NUMBER(10) REFERENCES STIL\_DE\_DANS(ID\_STIL),

ID\_SALA NUMBER(10) REFERENCES SALA(ID\_SALA));

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 1,100);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 1,200);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 1,300);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 2,400);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 2,500);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 3,600);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 4,700);

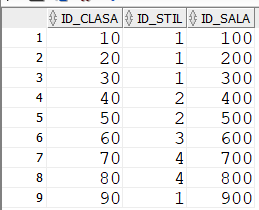
INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 4,800);

INSERT INTO CLASA VALUES

(GENERATOR.NEXTVAL, 1,900);

SELECT \* FROM CLASA;



CREATE TABLE CURSANT

(ID\_CURSANT NUMBER(10)PRIMARY KEY,NUME VARCHAR(20) NOT NULL,

PRENUME VARCHAR(20) NOT NULL,DATA\_NASTERII DATE,

EMAIL VARCHAR(20),

ID\_CLASA NUMBER(10) REFERENCES CLASA(ID\_CLASA));

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'POPESCU','MIHAELA',SYSDATE,NULL,10);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'GEORGESCU','IOANA',SYSDATE,'IOANA@GMAIL.COM'10);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'RUS','ALEXANDRU',SYSDATE,NULL,10);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'MECHIE','DARIA',SYSDATE,'JMECHIE@GMAIL.COM',10);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'TRANDAFIR','PATRICIA',SYSDATE,'BLUEPATY@POLI.COM',20);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'DUMITRU','MARIA',SYSDATE,'ESCALIBUR@CRAZY.COM',20);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'BONDOC','ANA',SYSDATE,NULL,20);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'OPREA','GEORGE',SYSDATE,NULL,30);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'TUDORACHE','MIRUNA',SYSDATE,NULL,30);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'PARASCHIV','STEFAN',SYSDATE,NULL,30);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'IVANCIUC','GHEORGHE',SYSDATE,NULL,30);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'DICULESCU','LUCA',SYSDATE,NULL,30);

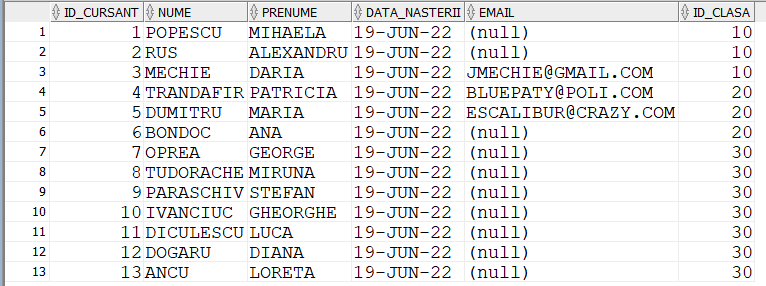
INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'DOGARU','DIANA',SYSDATE,NULL,30);

INSERT INTO CURSANT VALUES

(GENERATOR.NEXTVAL, 'ANCU','LORETA',SYSDATE,NULL,30);

SELECT \* FROM CURSANT;



CREATE TABLE EVENIMENT

(ID\_EVENIMENT NUMBER(10) PRIMARY KEY, DATA DATE , NRPUNCTE1 NUMBER(10),

NRPUNCTE2 NUMBER(10), NRPUNCTE3 NUMBER(10), NRPUNCTEPARTICIPARE NUMBER(10),

ID\_SEDIU NUMBER(10) REFERENCES SEDIU(ID\_SEDIU));

INSERT INTO EVENIMENT VALUES

(GENERATOR.NEXTVAL,SYSDATE, 20,15,10,5,3);

INSERT INTO EVENIMENT VALUES

(GENERATOR.NEXTVAL,SYSDATE, 30,10,5,0,1);

INSERT INTO EVENIMENT VALUES

(GENERATOR.NEXTVAL,SYSDATE, 50,25,10,5,2);

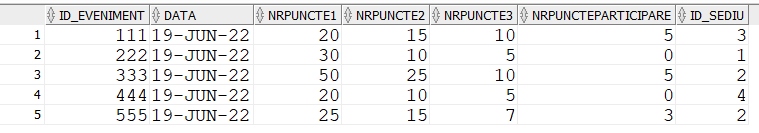
INSERT INTO EVENIMENT VALUES

(GENERATOR.NEXTVAL,SYSDATE, 20,10,5,0,4);

INSERT INTO EVENIMENT VALUES

(GENERATOR.NEXTVAL,SYSDATE, 25,15,7,3,2);

SELECT \* FROM EVENIMENT;



CREATE TABLE INCASARI(ID\_INCASARI NUMBER(20) PRIMARY KEY,

SUMA NUMBER(20), LUNA VARCHAR(20),

ID\_SEDIU NUMBER(10) REFERENCES SEDIU(ID\_SEDIU));

INSERT INTO INCASARI VALUES

(GENERATOR.NEXTVAL, 10000, 'SEPTEMBER', 1);

INSERT INTO INCASARI VALUES

(GENERATOR.NEXTVAL, 9550, 'OCTOBER', 1);

INSERT INTO INCASARI VALUES

(GENERATOR.NEXTVAL, 9000, 'NOVEMBER', 1);

INSERT INTO INCASARI VALUES

(GENERATOR.NEXTVAL, 5000, 'DECEMBER', 1);

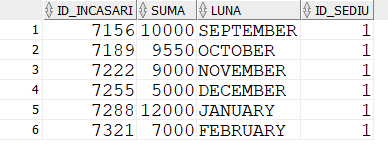
INSERT INTO INCASARI VALUES

(GENERATOR.NEXTVAL, 12000, 'JANUARY', 1);

INSERT INTO INCASARI VALUES

(GENERATOR.NEXTVAL, 7000, 'FEBRUARY', 1);

SELECT \* FROM INCASARI;



CREATE TABLE ANGAJAT

(ID\_SALARIAT NUMBER(10) PRIMARY KEY, NUME VARCHAR(20) NOT NULL,

PRENUME VARCHAR(20) NOT NULL , TELEFON VARCHAR(20));

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'GEORGE', 'MIHAI', '07232183');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'POPA', 'IOANA', '07273722');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'LUCULET', 'DORINA', '071112183');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'VASILE', 'CRISTIAN', '07232555');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'PARLOG', 'FILIP', '07236783');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'TANASE', 'DANIEL', '07192183');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'DUNCA', 'IOANA', '071976323');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'RADULESCU', 'DENISA', '07197773');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'OPREA', 'TUDOR', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'RUS', 'ION', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'MARIA', 'ANA', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'IONESCU', 'OANA', '075678911');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'GHEORGHE', 'ALEXANDRA', '073456113');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'PAUN', 'DARIA', '071223383');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'TOMA', 'MARIUS', '07987653');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'FLORINACHE', 'MATEI', '071122093');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'POP', 'PAUL', '076767673');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'MADGEARU', 'RADU', '07213412873');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'ION', 'RALUCA', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'IGNAT', 'SIMION', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'CONDRATOV', 'COSMIN', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'MACOVEI', 'ANA', '07144483');

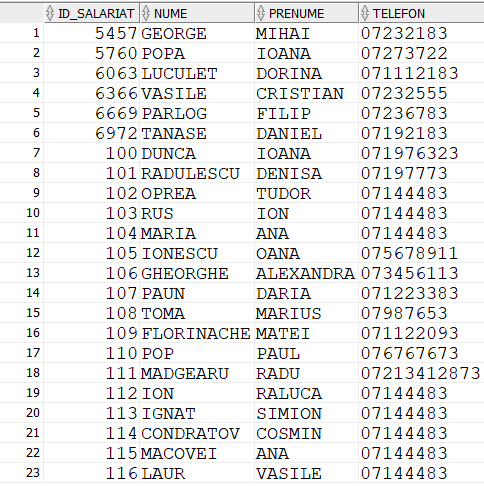
INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'LAUR', 'VASILE', '07144483');

INSERT INTO ANGAJAT VALUES

(GENERATOR.NEXTVAL, 'ivan', 'corina', '07654483');

SELECT \* FROM ANGAJAT;



CREATE TABLE INSTRUCTOR

(ID\_SALARIAT NUMBER(10) REFERENCES ANGAJAT(ID\_SALARIAT) PRIMARY KEY,

ID\_STIL NUMBER(10) REFERENCES STIL\_DE\_DANS(ID\_STIL));

INSERT INTO INSTRUCTOR VALUES (5760,1);

INSERT INTO INSTRUCTOR VALUES (5457,2);

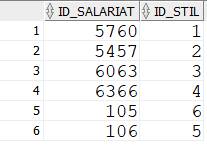
INSERT INTO INSTRUCTOR VALUES (6063,3);

INSERT INTO INSTRUCTOR VALUES (6366,4);

INSERT INTO INSTRUCTOR VALUES (105,6);

INSERT INTO INSTRUCTOR VALUES (106,5);

SELECT \* FROM INSTRUCTOR;



CREATE TABLE RECEPTIONIST(ID\_SALARIAT NUMBER(10) REFERENCES ANGAJAT(ID\_SALARIAT) PRIMARY KEY,

ID\_SEDIU NUMBER(10) REFERENCES SEDIU(ID\_SEDIU));

INSERT INTO RECEPTIONIST VALUES(6669,3);

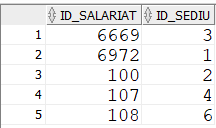
INSERT INTO RECEPTIONIST VALUES(6972,1);

INSERT INTO RECEPTIONIST VALUES(100,2);

INSERT INTO RECEPTIONIST VALUES(107,4);

INSERT INTO RECEPTIONIST VALUES(108,6);

SELECT \* FROM RECEPTIONIST;



CREATE TABLE PAZNIC(ID\_SALARIAT NUMBER(10) REFERENCES ANGAJAT(ID\_SALARIAT),

ID\_SEDIU NUMBER(10) REFERENCES SEDIU(ID\_SEDIU));

INSERT INTO PAZNIC VALUES(101,1);

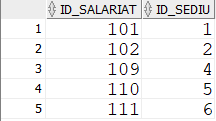
INSERT INTO PAZNIC VALUES(102,2);

INSERT INTO PAZNIC VALUES(109,4);

INSERT INTO PAZNIC VALUES(110,5);

INSERT INTO PAZNIC VALUES(111,6);

SELECT \* FROM PAZNIC;



CREATE TABLE ANGAJAT\_CURATENIE (ID\_SALARIAT NUMBER(10) REFERENCES ANGAJAT(ID\_SALARIAT),

ID\_SEDIU NUMBER(10) REFERENCES SEDIU(ID\_SEDIU));

INSERT INTO ANGAJAT\_CURATENIE VALUES(103,1);

INSERT INTO ANGAJAT\_CURATENIE VALUES (104,2);

INSERT INTO ANGAJAT\_CURATENIE VALUES (112,1);

INSERT INTO ANGAJAT\_CURATENIE VALUES (113,3);

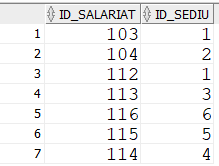
INSERT INTO ANGAJAT\_CURATENIE VALUES (116,6);

INSERT INTO ANGAJAT\_CURATENIE VALUES (115,5);

INSERT INTO ANGAJAT\_CURATENIE VALUES (114,4);

insert into paznic values(14,3);

SELECT \* FROM ANGAJAT\_CURATENIE;



CREATE TABLE ORA\_DE\_CURS

(ID\_ORA NUMBER(10) PRIMARY KEY, DATA DATE, ORA\_INCEPUT VARCHAR(20),

DURATA NUMBER(10), ID\_STIL NUMBER(10) REFERENCES STIL\_DE\_DANS(ID\_STIL),

ID\_SALARIAT NUMBER(10) REFERENCES ANGAJAT(ID\_SALARIAT));

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'18:00',90,1,6063);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'20:00',90,1,6063);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'18:00',90,2,5760);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'18:00',90,3,5457);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'18:00',90,4,6366);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'20:00',90,4,6366);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'18:00',90,4,105);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'19:45',90,4,105);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'17:00',90,6,106);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'18:45',90,4,106);

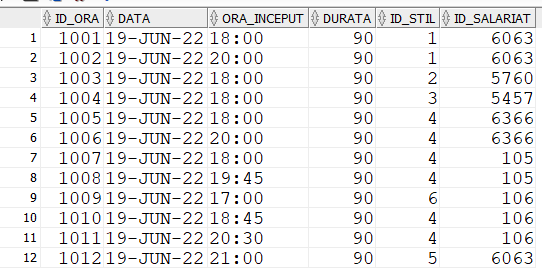
INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'20:30',90,4,106);

INSERT INTO ORA\_DE\_CURS VALUES

(GENERATOR.NEXTVAL, SYSDATE ,'21:00',90,5,6063);

SELECT \* FROM ORA\_DE\_CURS;



*11.Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente: •operație join pe cel puțin 4 tabele•filtrare la nivel delinii•subcereri sincronizateîn care intervin cel puțin 3 tabele•subcereri nesincronizateîn care intervin cel puțin 3 tabele•grupări de date, funcții grup, filtrare la nivel degrupuri•ordonări•utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE•utilizarea a cel puțin 1 bloc de cerere(clauza WITH).*

**Cerinta 1: Sa se afiseze denumirea stilului de dans predat la ora 18:00 din data curenta si instructorul care preda acel stil de dans la ora specificata.**

--EX1 denumirea stilului de dans predat la ora 18:00 din data 19.06.2022 si instructorul care il preda

select s.denumire, i.id\_salariat,o.data from stil\_de\_dans s, ora\_de\_curs o,instructor i

where s.id\_stil=o.id\_stil and o.id\_stil=i.id\_stil and o.ora\_inceput='18:00' and TRUNC(o.data)=to\_date('19-JUN-2022', 'dd-mm-YYYY');

**CERINTA 2: Sa se afiseze numele si prenumele cursantilor care au cel mai scump abonament**

select c.nume, c.prenume, a.tip from cursant c, abonament a

where a.pret=(select max(pret) from abonament);

**CERINTA 3: Un sediu este MARE daca are peste 7 sali, MEDIU daca are peste 4 sali si MIC daca are sub 3 sali. Administratorul vrea sa stie daca un sediu se incadreaza la categoria MARE, MEDIU sau MIC.**

**Sa se afiseze sediile si categoria din care fac parte.**

SELECT ID\_SEDIU,

CASE

WHEN NR\_SALI>=7 THEN 'MARE'

WHEN NR\_SALI<=7 AND NR\_SALI>=4 THEN 'MEDIU'

ELSE 'MIC'

END AS CATEGORIE

FROM SEDIU;

**CERINTA 4: Sa se afiseze incasarile din fiecare luna in ordine crescatoare**

with minc as (select luna, sum(suma) as thing

from incasari group by luna order by sum(suma) asc)

select \* from minc;

**CERINTA 5: Sa se afiseze id-ul instructorilor si denumirea stilului de dans pe care il predau**

select id\_salariat,

decode(id\_stil,1,'DANCEHALL',

2,'HIP HOP',

3,'LOCKING',

4,'HOUSE',

5,'WAACKING',

6,'VOGUE',

NULL) STIL\_PREDAT

FROM INSTRUCTOR;

**Cerinta 6: Sa se afiseze pentru cursantii care nu au email mesajul ‚nu are email’**

select id\_cursant, NVL(TO\_CHAR(email),'NU are email') from cursant;

**Cerinta 7: sa se afiseze evenimentele care ofera un numar de puncte pentru locul 3 care incepe cu cifra 1**

select to\_Char(nrpuncte3) from eveniment

where to\_Char(nrpuncte3) like '1%';

*12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.*

Operatia 1: delete

Stergem paznicul de la sediul cu cele mai putine sali.

**delete from paznic p**

**where p.id\_sediu=(select id\_sediu from sediu where nr\_sali=(select min(nr\_sali)from sediu));**

Operatia 2: update

Mutam clasa cu cel mai mare id intr o sala dintr-un alt sediu

**update clasa set**

**id\_sala=(select min(id\_sala) from sala where id\_sediu=(select min(id\_sediu)from sediu))**

**where id\_clasa=(select max(id\_clasa)from clasa);**

Operatia 3: delete

Stergem un sediu aflat la o adresa din cel mai mare sector

**delete from adresa s**

**where s.sector=(select max(sector) from adresa);**

1. *Crearea unei vizualizari compuse. Dati UN exemplu de operatie LMD permisa pe vizualizarea respectiva si un exemplu de operatie nepermisa.*

Prima condiţie ca o vizualizare a cărei cerere conţine un join să fie modificabilă este ca instrucţiunea LMD să afecteze un singur tabel din operaţia de join.

Vizualizarea compusa:

**Create view vizualizare (id\_cursant, nume, prenume, data\_nasterii,id\_clasa,id\_stil,id\_sala)**

**as select c.id\_cursant, c.nume, c.prenume, c.data\_nasterii, c.id\_clasa, cl.id\_stil, cl.id\_sala**

**from cursant c, clasa cl where c.id\_clasa=cl.id\_clasa;**

Exemplu de operatie LMD nepermisa:

**insert into vizualizare values (23,'ana','maria',sysdate,555,10,20);**

*nu functioneaza deoarece incercam sa introducem valori in mai multe tabele deodata*

Exemplu de operatie LMD permisa:

**delete from vizualizare where id\_cursant<=3;**

*16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.*

--se vor afisa salile in care se desfasoare ore de curs, precum si clasele participante

--cursantii din clase

--sediile unde se afla salile

select c.id\_cursant,cl.id\_Clasa,s.id\_sala,se.id\_sediu

from (((cursant c full outer join clasa cl on c.id\_clasa=cl.id\_clasa) full outer join sala s on s.id\_sala=cl.id\_sala)

full outer join sediu se on s.id\_sediu=se.id\_sediu)

order by c.id\_clasa;

--division

--sa se afiseze numele cursantilor care se afla in clasa 20

select nume, prenume from cursant where id\_clasa in (select id\_clasa from clasa where id\_clasa=20)

group by nume,prenume

having count(id\_clasa)=(select count(\*) from clasa where id\_clasa=20);

--sa se afiseze id ul evenimentelor care s-au desfasurat in sediul cu id ul 2

select distinct id\_eveniment from eveniment

where not exists (select 1 from sediu s where s.id\_sediu=6 and not exists (select 'x' from eveniment e where e.id\_sediu=s.id\_sediu));

*17. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile*

*operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore*

*algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.*

Vom face o cerere care furnizeaza informatii despre ora de curs in care se preda Dancehall

Cererea neoptimizata:

select o.\* from ora\_de\_curs o, stil\_de\_dans s

where o.id\_stil=s.id\_stil and lower(s.denumire)='dancehall';

0.016s

Cererea opimizata:

select o.\* from ora\_de\_curs o, stil\_de\_dans s

where lower(s.denumire)='dancehall' and o.id\_stil=s.id\_stil;

0.002s

ARBORE OPTIMIZAT

R1=JOIN(ora\_de\_curs, stil)

R2=SELECT(R1, o.id\_stil=s.id\_stil)

REZULTAT=SELECT(R2, ,s.denumire=’Dancehall’)

STIL DE DANS

ORA DE CURS

s.denumire=’Dancehall’

REZULTAT

o.id\_stil=s.id\_stil