

Universitatea din București
Facultatea de Matematică și Informatică
Domeniul Tehnologia Informației

**Proiect la disciplina
BAZE DE DATE**

Profesor Coordonator:
Silviu Laurențiu Vasile

Student:
Vlaicu Ana-Francesca
Anul II, grupa 263

BUCUREȘTI
2023

Universitatea din București
Facultatea de Matematică și Informatică
Domeniul Tehnologia Informației

Studio de dans

Profesor Coordonator:
Silviu Laurențiu Vasile

Student:
Vlaicu Ana-Francesca
Anul II, grupa 263

BUCUREȘTI
2023

1. Prezentare:

Tema aleasă este „STUDIO DE DANS” și presupune crearea unei baze de date folosită pentru administrarea unui studio de dans.

Studio-ul oferă pasionaților de dans oportunitatea de a învăța diverse stiluri de dans, de la cele Street (Hip Hop, Popping, etc), la dansuri moderne și contemporane și inclusiv stiluri africane, jamaicane, invitând potențiali elevi să-și dezvolte pasiunile sau chiar să primească o diplomă de performanță. Locațiile studio-ului au toate dotările necesare pentru a putea fi închiriate de dansatori sau trupe de dans pentru repetiții sau evenimente private, precum workshops, petreceri etc.

Pentru o bună funcționare a studio-ului, baza de date prin care se administrează datele trebuie să fie una integră, bine definită și care să producă foarte puține sau chiar 0 erori.

2. Reguli:

Elevi:

- Un elev poate avea mai multe înscrieri.
- Un elev poate susține unul sau mai multe examene (poate da examenul din nou în cazul în care nu promovează din prima).

Înscrieri:

- O înscriere aparține unui elev.
- O înscriere se creează pentru un curs.
- O înscriere este achitată printr-o plată.

Cursuri:

- Pentru un curs se pot crea mai multe înscrieri.
- Un curs trebuie condus de un instructor.
- Un curs se desfășoară la o singură locație.

Locații:

- La o locație se desfășoară cel puțin un curs.
- La o locație se pot rezerva mai multe închirieri.

Plăți:

- O plată este executată pentru o înscriere.

Examene:

- Un examen poate fi susținut de un singur elev.

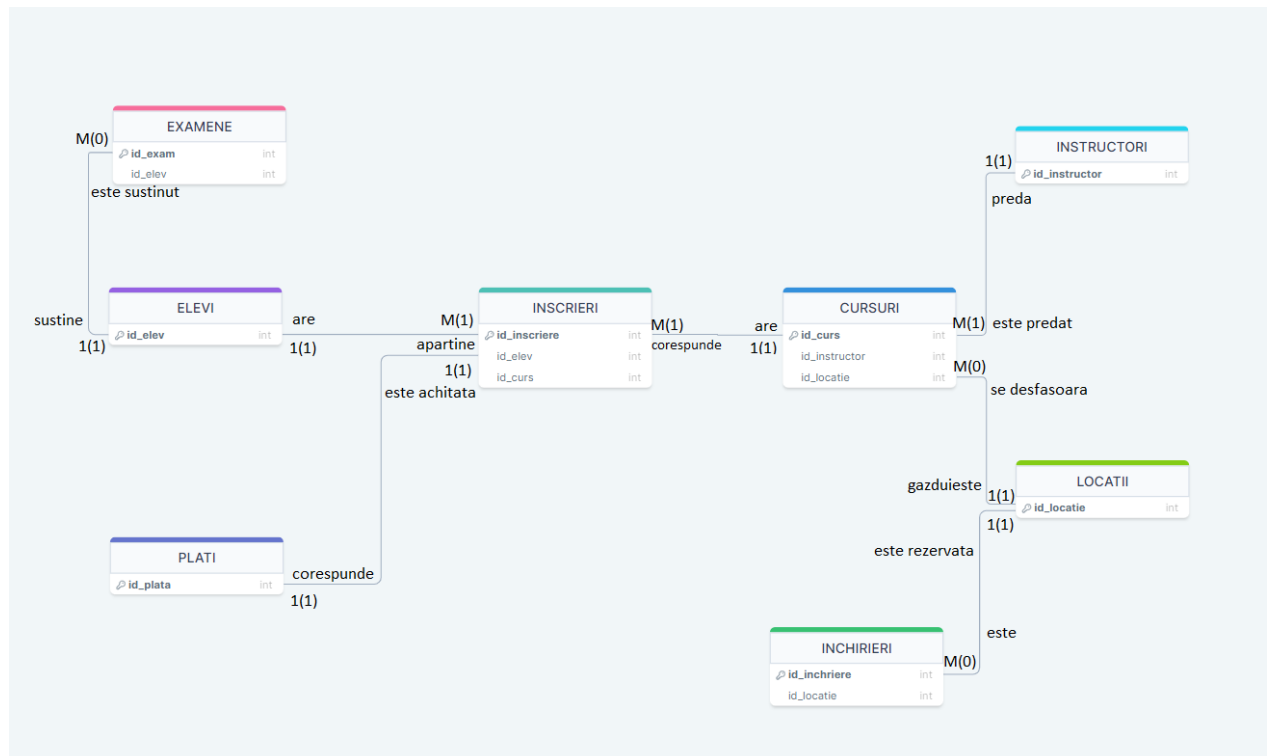
Instructori:

- Un instructor poate predă mai multe cursuri.

Închirieri:

- O închiriere se poate desfășura la o singură locație

3. Diagrama entitate-relație:



3.1 Descrierea tabelelor, entităților și a atributelor:

Elevi:

- Tabela Elevi a fost creată pentru a ține evidența elevilor care se înscriu la activitățile studio-ului de dans. Ea este compusă din cheia primară *id_elev* de tip NUMBER și cinci atribute:
 - *nume*: înregistrează numele de familie al fiecărui elev; este de tip VARCHAR2 și are o lungime maximă de 30 de caractere; nu poate lipsi din rândul unui elev.
 - *prenume*: înregistrează prenumele fiecărui elev; este de tip VARCHAR2 și are o lungime maximă de 30 de caractere.
 - *email*: înregistrează adresa de email a fiecărui elev; este de tip VARCHAR2 și are o lungime maximă de 40 de caractere.
 - *nr_telefon*: înregistrează numărul de telefon al elevului; este de tip VARCHAR2 și are lungimea maximă de 12 caractere; nu poate lipsi din înregistrarea unui elev.

- *data_nastere*: înregistrează data de naștere a elevului; este de tip DATE.

Examene:

- Tabela Examene a fost creată pentru a ține evidența examenelor date de elevi. Elevii nu sunt obligați să participe la examene, acestea deservind doar celor care vor să obțină o diplomă de absolvire a cursurilor unei școli de dans. Practic, un elev se poate înscrie în cadrul studio-ului pentru propria sa distracție sau pentru a primi o certificare. Examenele sunt individuale pentru elevi; sunt open style, urmărind aspecte predate la fiecare curs precum muzicalitate, coordonare, tehnici, stamina în defavoarea specializării doar pe un stil/curs anume.
- Tabela Examene conține cheia primară *id_exam* de tip NUMBER, cheia străină *id_elev* ce face referință la tabela ELEVI și alte două atribute:
 - *data_exam*: înregistrează data la care un elev a susținut un examen; este de tip DATE.
 - *rezultat*: înregistrează rezultatul elevului la examenul susținut de el; este de tip VARCHAR2 și are o lungime maximă de 8 caractere; intrările în coloana rezultat pot fi ‘promovat’, ‘picat’ sau null (null pentru situația în care examenul a fost programat și nu a fost susținut încă sau dacă nu a mai fost susținut din varii motive).

Instructori:

- Tabela instructori conține informații despre instructori care predau în cadrul studio-ului. Aceasta are cheia primară *id_instructor* de tip NUMBER și următoarele atribute:
 - *nume*: reține numele de familie al instructorului; este de tip VARCHAR2 și are o lungime maximă de 30 de caractere; trebuie adăugat obligatoriu.
 - *prenume*: înregistrează prenumele instructorului; este de tip VARCHAR2 și are o lungime maximă de 30 de caractere;
 - *calificare*: înregistrează aria de dans/stilurile în care este dezvoltat instructorul; este de tip VARCHAR2 și are o lungime maximă de 20 de caractere; trebuie adăugat obligatoriu.
 - *nr_telefon*: înregistrează numărul de telefon al instructorului; este de tip VARCHAR2 și are lungimea maximă de 12 caractere; trebuie adăugat obligatoriu.
 - *email*: înregistrează adresa de email a instructorului este de tip VARCHAR2 și are o lungime maximă de 40 de caractere.

Locații:

- Tabela Locații conține toate locațiile la care studio-ul de dans desfășoară activități. Cheia primară este *id_locație*, de tip NUMBER iar celelalte attribute sunt:
 - *adresa*: înregistrează strada și numărul la care se află o locație; este de tip VARCHAR2 și are o lungime maximă de 30 de caractere; trebuie adăugată obligatoriu.
 - *capacitate*: înregistrează numărul maxim de persoane care pot participa la activități într-o locație în același timp; este de tip NUMBER și are o lungime maximă de 2 cifre.

Închirieri:

- Tabela Închirieri ține evidența tuturor închirierilor de la fiecare locație. Studio-ul de dans oferă posibilitatea de a închiria spațiile de care dispune oricărei persoane care dorește să susțină repetiții private, workshop-uri etc într-un spațiu echipat pentru dans. Tabela are o cheie primară *id_inchiriere* de tip NUMBER și o cheie străină *id_locatie*, cheia primară a tablei Locații. Celelalte attribute sunt:
 - *nume_client*: înregistrează numele clientului care închiriază una dintre locațiile studioului; este de tip VARCHAR2 și are o lungime maximă de 30 de caractere.
 - *nr_telefon*: înregistrează numărul de telefon al clientului; este de tip VARCHAR2 și are lungimea maximă de 12 caractere.
 - *data_inchiriere*: înregistrează data în care spațiul este închiriat clientului; este de tip DATE.
 - *ora_inceput*: înregistrează ora de la care spațiul este în folosința clientului; este de tip VARCHAR2 și are lungimea maximă de 5 caractere; ora se va scrie în format 'HH24:MI'.
 - *ora_sfarsit*: înregistrează ora până la care spațiul este în folosința clientului; este de tip VARCHAR2 și are lungimea maximă de 5 caractere; ora se va scrie în format 'HH24:MI'.
 - *cost_inchiriere*: înregistrează suma pe care o va plăti clientul pentru închirierea spațiului; costul nu este fix, el depinde de mai multe aspecte, precum numărul de ore, capacitatea spațiului, motivul închirierii etc, astfel că el se introduce manual (în urma unei înțelegeri între client și studio) în baza de date și nu se poate calcula automat; este de tip NUMBER și are o lungime de maxim 4 cifre.
- Toate attributele trebuie adăugate la fiecare intrare obligatoriu.

Cursuri:

- Tabela Cursuri ține evidența cursurilor organizate de studio. Cursurile se desfășoară săptămânal, în aceeași zi, de la data de început până la data de sfârșit (ca un fel de modul). Cheia primară este *id_curs*, de tip NUMBER, iar cheile străine sunt *id_instructor*, făcând legătura cu instructorul care ține cursul și *id_locatie*, indicând locația la care se desfășoară cursul. Celelalte atribute sunt:
 - *nume_curs*: înregistrează numele cursului; este de tip VARCHAR2 și are o lungime maximă de 20 de caractere; trebuie adăugat obligatoriu.
 - *nivel*: indică nivelul de dificultate al cursului; acesta nu condiționează participarea unui elev la curs, ci are o valoare orientativă, astfel încât cei care se înscriu la acest curs să aibă o idee despre greutatea acestuia; este de tip VARCHAR2 și are o lungime de maxim 10 caractere; poate fi null, deoarece un curs poate fi open level (destinat oricui).
 - *ora_inceput*: înregistrează ora la care începe cursul; este de tip VARCHAR2 și are lungimea maximă de 5 caractere; trebuie adăugată obligatoriu; ora se va scrie în format 'HH24:MI'.
 - *ora_sfarsit*: înregistrează ora la care se termină cursul; este de tip VARCHAR2 și are lungimea maximă de 5 caractere; trebuie adăugată obligatoriu; ora se va scrie în format 'HH24:MI'.
 - *data_inceput*: înregistrează data de la care începe cursul; este de tip DATE; trebuie adăugată obligatoriu.
 - *data_sfarsit*: înregistrează data până la care se ține cursul; este de tip DATE.
 - *zi_sapt*: înregistrează ziua din săptămână în care se desfășoară cursul; este de tip VARCHAR2 și are o lungime de maxim 10 caractere; trebuie adăugată obligatoriu.
 - *cost_curs*: înregistrează prețul integral al cursului; este de tip NUMBER și are maxim 4 cifre.

Înscrieri:

- Tabela Înscrieri contorizează înscrierile elevilor la cursuri. Aceasta are cheia primară *id_inscriere* de tip NUMBER, cheile străine *id_curs* și *id_elev* ce fac legătura cu tabelele Elevi, respectiv Cursuri și un atribut:
 - *data_inscriere*: înregistrează data la care un elev s-a înscris la un curs; este de tip DATE; trebuie adăugată obligatoriu.

Plăți:

- Tabela Plăți ține evidența plăților pe care elevii le fac pentru a putea participa la cursurile organizate de studio. Are cheia primară de tip NUMBER *id_plata* ce face referință la *id_inscriere*. Celelalte attribute sunt:
 - *metoda_de_plata*: înregistrează modul în care a fost plătită înscrierea de elev; este de tip VARCHAR2 și de lungime maximă de 10 caractere; trebuie adăugată obligatoriu.
 - *suma*: înregistrează suma pe care a plătit-o un elev pentru o singură înscriere; este de tip NUMBER și de maxim 4 cifre; trebuie adăugată obligatoriu.
 - *data_plata*: înregistrează data la care s-a efectuat plata; este de tip DATE; nu poate fi nulă.

3.2 Descrierea relațiilor și a cardinalităților:

Elevi – Înscrieri:

Relație: one-to-many

- Un elev trebuie să aibă înscrieri.
- O înscriere trebuie să aparțină unui elev.
- Un elev are una sau mai multe înscrieri.
- O înscriere aparține unui singur elev.

Elevi – Examene:

Relație: one-to-many

- Un elev poate susține examene.
- Un examen trebuie să fie susținut de un elev.
- Un elev susține unul sau mai multe examene.
- Un examen este susținut doar de un elev .

Cursuri – Instructori:

Relație: many-to-one

- Un curs trebuie predat de un instructor.
- Un instructor trebuie să predea cursuri.
- Un curs este predat de un instructor.
- Un instructor predă unul sau mai multe cursuri.

Cursuri – Înscrieri:

Relație: one-to-many

- Un curs trebuie să aibă cel puțin o înscriere.
- O înscriere trebuie să corespundă unui curs.
- Un curs are una sau mai multe înscrieri.
- O înscriere corespunde unui singur curs.

Cursuri – Locații:

Relație: many-to-one

- Un curs trebuie să se desfășoare la o locație.
- O locație poate găzdui cursuri.
- Un curs se desfășoară la o locație.
- O locație găzduiește mai multe cursuri.

Plăți – Înscrieri:

Relație: one-to-one

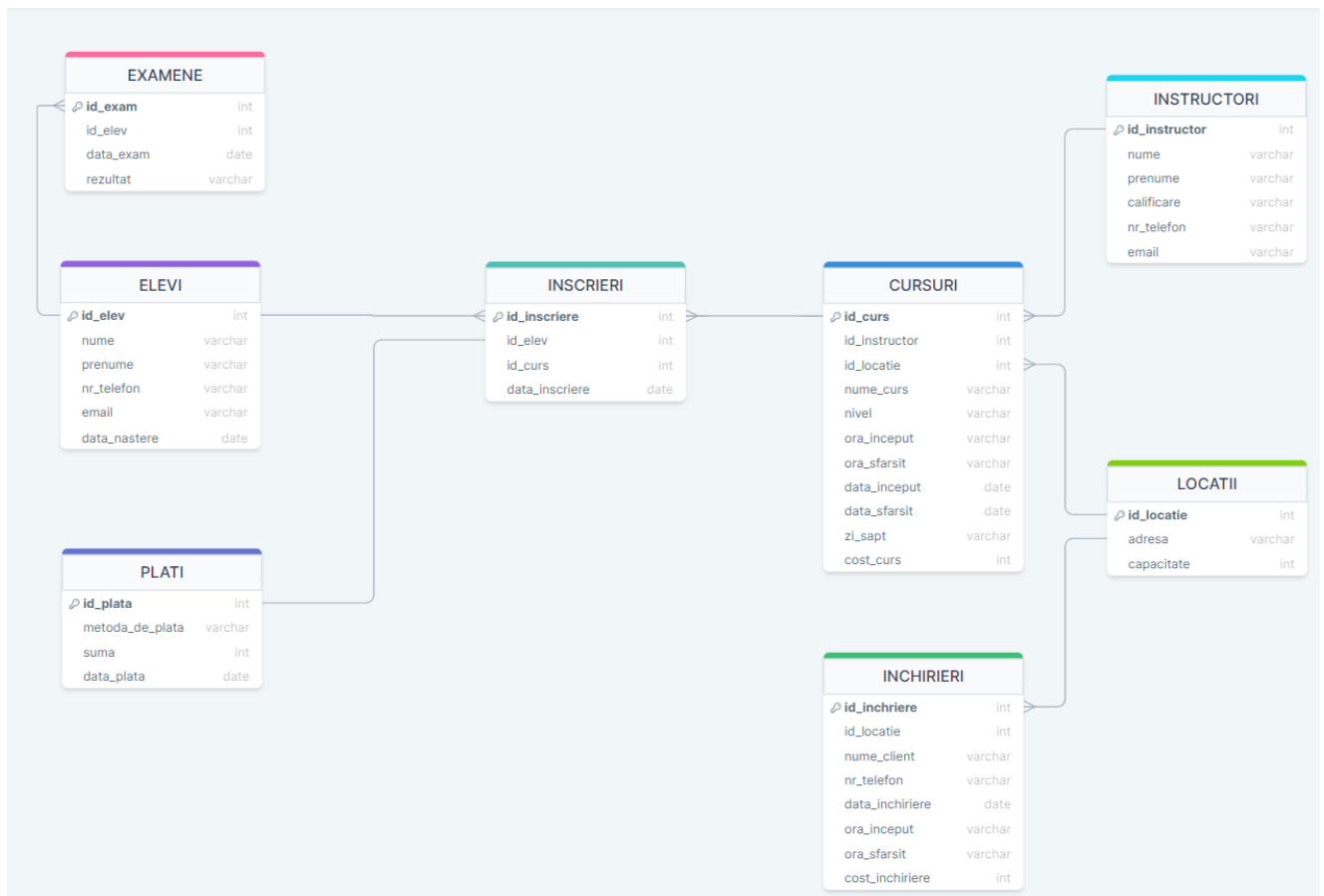
- O plată trebuie să corespundă unei înscrieri.
- O înscriere trebuie achitată printr-o plată.
- O plată corespunde unei singure înscrieri.
- O înscriere este achitată printr-o singură plată.

Locații – Închirieri:

Relație: one-to-many

- O locație poate fi rezervată pentru închirieri.
- O închiriere trebuie să fie la o locație.
- O închiriere este la o singură locație.
- O locație este rezervată pentru una sau mai multe închirieri.

4. Diagrama conceptuală:



4.1 Descrierea constrângerilor de integritate:

Instructori:

- Constrângere de tip PRIMARY KEY: ***pk_instructori*** pentru atributul *id_instructor*.
- Constrângeri de tip NOT NULL: pentru attributele *nume*, *calificare*, *nr_telefon*, întrucât aceste attribute sunt obligatorii.

Locații:

- Constrângere de tip PRIMARY KEY: ***pk_locatie*** pentru atributul *id_locatie*.
- Constrângeri de tip NOT NULL: pentru attributele *adresa*, *capacitate*, întrucât aceste attribute sunt obligatorii.

Închirieri:

- Constrângere de tip PRIMARY KEY: ***pk_inchirieri*** pentru atributul *id_inchiriere*.
- Constrângere de tip FOREIGN KEY: ***fk_id_loc_inchr*** pentru atributul *id_locatie*; face legătură cu tabela Locații și indică locația care a fost închiriată; Pentru cheia străină, este creată o constrângere ON DELETE CASCADE, astfel încât, atunci când se șterge o locație, să se șteargă și închirierile de la locația respectivă.
- Constrângeri de tip NOT NULL: pentru attributele *id_locatie*, *nume_client*, *nr_telefon*, *data_inchiriere*, *ora_inceput*, *ora_sfarsit*, deoarece aceste date trebuie introduce obligatoriu.

Cursuri:

- Constrângere de tip PRIMARY KEY: ***pk_cursuri*** pentru atributul *id_curs*.
- Constrângeri de tip FOREIGN KEY: ***fk_id_locatie_curs*** pentru atributul *id_locatie*; face legătură cu tabela Locații și indică locația unde se desfășoară cursul; Pentru cheia străină, este creată o constrângere ON DELETE CASCADE, astfel încât, atunci când se șterge o locație, să se șteargă și cursurile care aveau loc la aceasta; de asemenea, ***fk_id_instructor*** pentru atributul *id_instructor*; face legătură cu tabela Instructori și indică instructorul care desfășoară cursul; Pentru cheia străină, este creată o constrângere ON DELETE CASCADE, astfel încât, atunci când se șterge un instructor, să se șteargă și cursurile pe care acesta le predă.
- Constrângere de tip CHECK: ***chk_date_curs*** pentru a compara attributele *data_inceput*, *data_sfarsit*, astfel încât data la care începe cursul să fie întotdeauna înaintea celei la care se termină cursul.

- Constrângeri de tip NOT NULL: pentru attributele *id_locatie*, *id_instructor*, *nume_curs*, *ora_inceput*, *ora_sfarsit*, *data_inceput*, *data_sfarsit*, *zi_sapt*, deoarece aceste attribute trebuie introduce obligatoriu.

Elevi:

- Constrângere de tip PRIMARY KEY: ***pk_elevi*** pentru atributul *id_elev*.
- Constrângeri de tip NOT NULL: pentru attributele *nume*, *nr_telefon*, deoarece aceste attribute sunt obligatorii.

Examenene:

- Constrângere de tip PRIMARY KEY: ***pk_exami*** pentru atributul *id_exam*.
- Constrângeri de tip FOREIGN KEY: ***fk_id_elevi_exam*** pentru atributul *id_elevi*; face legătură cu tabela Elevi și indică elevul care a susținut examenul; Pentru cheia străină, este creată o constrângere ON DELETE CASCADE, astfel încât, atunci când se șterge o un elev, să se șteargă și datele examenelor susținute de acesta.
- Constrângere de tip CHECK: ***chk_rezultat*** pentru a putea introduce în coloana *rezultat* doar calificativele ‘promovat’ sau ‘picat’.
- Constrângeri de tip NOT NULL: pentru attributele *id_elev*, *data_exam*, întrucât aceste date trebuie introduce obligatoriu.

Înscrieri:

- Constrângere de tip PRIMARY KEY: ***pk_inscrieri*** pentru atributul *id_inscriere*.
- Constrângeri de tip FOREIGN KEY: ***fk_id_elevi_inscr*** pentru atributul *id_elevi*; face legătură cu tabela Elevi și indică elevul căruia îi aparține înscrierea; Pentru cheia străină, este creată o constrângere ON DELETE CASCADE, astfel încât, atunci când se șterge un elev, să i se șteargă înscrierea; de asemenea, ***fk_id_curs_inscr*** pentru atributul *id_curs*; face legătură cu tabela Cursuri și arată cursul pentru care s-a făcut înscrierea; Pentru cheia străină, este creată o constrângere ON DELETE CASCADE, astfel încât, atunci când se șterge un curs, să se șteargă și înscrierile care s-au făcut pentru acesta.
- Constrângeri de tip NOT NULL: pentru attributele *id_elev*, *id_curs*, *data_inscriere*.
-

Plăți:

- Constrângere de tip PRIMARY KEY: *pk_plati* pentru atributul *id_plata*.
- Constrângeri de tip FOREIGN KEY: *pk_fk_id_plata* pentru atributul *id_plata*; face legătură cu tabela Înscrieri și indică înscrierea pentru care s-a făcut plata.
- Constrângeri de tip NOT NULL: pentru attributele *metoda_de_plata*, *suma*, *data_plata*.

4.2 Scheme relaționale:

- ELEVI(id_elev#, nume, prenume, nr_telefon, email, data_nastere);
- EXAMENE(id_exam#, id_elev(fk), data_exam, rezultat);
- PLATI(id_plata#(pk, fk), metoda_de_plata, data_plata);
- INSTRUMENTORI(id_instructor#, nume, prenume, calificare, nr_telefon, email);
- LOCATII(id_locatie#, adresa, capacitate);
- CURSURI(id_curs#, id_instructor(fk), id_locatie(fk), nume_curs, nivel, ora_inceput, ora_sfarsit, data_inceput, data_sfarsit, zi_sapt, cost_curs);
- INCHIRIERI(id_inchiriere#, id_locatie(fk), nume_client, nr_telefon, data_inchiriere, ora_inceput, ora_sfarsit, cost_inchiriere);
- INSCRIERI(id_inscriere#, id_elev(fk), id_curs(fk), data_inscriere).

5. Crearea tabelor și inserarea de valori:

Baza de date a fost implementată în Oracle SQL Developer prin comanda CREATE TABLE. Constrângerile au fost adăugate prin cuvintele cheie ADD CONSTRAINT/CONSTRAINT. Am avut în vedere implementarea tabelor fără chei străine înaintea celor cu.

Instructori:

```
create table INSTRUCTORI(  
id_instructor number(4) constraint pk_instructori primary key,  
nume varchar2(30) not null,  
prenume varchar2(30),  
calificare varchar2(20) not null,  
nr_telefon varchar(12) not null,  
email varchar2(40));
```

Locații:

```
create table LOCATII(  
id_locatie number(4) constraint pk_locatie primary key,  
adresa varchar2(30) not null,  
capacitate number(2) not null);
```

Închirieri:

```
create table INCHIRIERI(  
id_inchiriere number(3) constraint pk_inchirieri primary key,  
id_locatie number(4) not null,  
nume_client varchar2(30) not null,  
nr_telefon varchar2(12) not null,  
data_inchiriere date not null,  
ora_inceput varchar2(5) not null,  
ora_sfarsit varchar2(5) not null,  
cost_inchiriere number(4),  
constraint fk_id_loc_inchr foreign key (id_locatie) references LOCATII(id_locatie) on delete cascade);
```

Cursuri:

```
create table CURSURI(  
id_curs number(3) constraint pk_cursuri primary key,  
id_instructor number(3) not null,  
id_locatie number(4) not null constraint fk_id_locatie_curs references LOCATII(id_locatie) on delete cascade,  
nume_curs varchar2(20) not null,  
nivel varchar2(10),  
ora_inceput varchar2(5) not null,  
ora_sfarsit varchar2(5) not null,  
data_inceput date not null,  
data_sfarsit date not null,  
zi_sapt varchar2(10) not null,  
cost_curs number(4),  
constraint fk_id_instructor foreign key (id_instructor) references INSTRUCTORI(id_instructor) on delete cascade,  
constraint chk_date_curs check (data_inceput < data_sfarsit));
```

Elevi:

```
create table ELEVI(  
id_elev number(3) constraint pk_elevi primary key,  
nume varchar2(30) not null,  
prenume varchar2(30),  
email varchar2(40),  
nr_telefon varchar2(12) not null,  
data_nastere date);
```

Examene:

```
create table EXAMENE(  
id_exam number(3) constraint pk_exam primary key,  
id_elev number(3) not null,  
data_exam date not null,  
rezultat varchar2(8) constraint chk_rezultat check (UPPER(rezultat) IN('PROMOVAT', 'PICAT')),  
constraint fk_id_elevi_exam foreign key (id_elev) references ELEVI(id_elev) on delete cascade);
```

Înscrieri:

```
create table ÎNSCRIERI(  
id_inscriere number(3) constraint pk_inscrieri primary key,  
id_elev number(3) not null,  
id_curs number(3) not null,  
data_inscriere date not null,  
constraint fk_id_elevi_inscr foreign key (id_elev) references ELEVI(id_elev) on delete cascade,  
constraint fk_id_curs_inscr foreign key (id_curs) references CURSURI(id_curs) on delete cascade);
```


Plăți:

Am folosit și comanda ALTER TABLE pentru a adăuga o constrângere după ce am creat tabela.

```
create table PLATI(  
id_plata number(3) constraint pk_plati primary key,  
metoda_de_plata varchar2(10) not null,  
suma number(4) not null,  
data_plata date not null);  
  
alter table PLATI add constraint pk_fk_id_plata foreign key (id_plata) references INSCRIERI(id_inscriere);
```

Pentru inserarea de valori, am folosit comanda INSERT INTO ... VALUES.

Elevi:

```
insert into ELEVI values(100, 'Popescu', 'Alexandru', 'popalex@gmail.com', '0722232323', to_date('12-04-2002','dd-mm-yyyy'));  
insert into ELEVI values(101, 'Bradea', 'Ioana', 'braioana@gmail.com', '0722232324', to_date('26-02-1999','dd-mm-yyyy'));  
insert into ELEVI values(102, 'Marin', 'Raluca', 'ralum@yahoo.com', '0722232325', to_date('04-11-2006','dd-mm-yyyy'));  
insert into ELEVI values(103, 'Voicu', 'Dan', 'danvoicu@gmail.com', '0722232326', to_date('12-11-2004','dd-mm-yyyy'));  
insert into ELEVI values(104, 'Negoiu', 'Vlad', 'vladn@gmail.com', '0722232327', to_date('30-03-2002','dd-mm-yyyy'));  
insert into ELEVI values(105, 'Neacsu', 'Andra', 'andra02@gmail.com', '0722232328', to_date('02-05-2002','dd-mm-yyyy'));  
insert into ELEVI values(106, 'Andrei', 'Rares', 'arares21@yahoo.com', '0722232329', to_date('14-10-2003','dd-mm-yyyy'));  
insert into ELEVI values(107, 'Enescu', 'Daria', 'dariaenescu@yahoo.com', '0722232330', to_date('14-12-2001','dd-mm-yyyy'));
```

Locații:

```
insert into LOCATII values(200, 'str. Unirii nr. 82', 80);  
insert into LOCATII values(201, 'str. Macului nr. 40', 90);  
insert into LOCATII values(202, 'str. Florilor nr. 9A', 75);  
insert into LOCATII values(203, 'str. Traian Vuia nr. 6', 80);
```

Examene:

```
insert into EXAMENE values(300, 103, to_date('01-10-2021','dd-mm-yyyy'), 'picat');  
insert into EXAMENE values(301, 105, to_date('05-10-2021','dd-mm-yyyy'), 'promovat');  
insert into EXAMENE values(302, 107, to_date('01-10-2021','dd-mm-yyyy'), 'promovat');  
insert into EXAMENE values(303, 104, to_date('05-10-2021','dd-mm-yyyy'), null);
```

Instructori:

```
insert into INSTRUCTORI values(400, 'Dumitru', 'Matei', 'streetdance', '0722111110', 'dumatei@yahoo.com');
insert into INSTRUCTORI values(401, 'Mita', 'Bogdan', 'dans modern', '0722111111', 'mbogdan9@yahoo.com');
insert into INSTRUCTORI values(402, 'Gorja', 'Emma', 'streetdance', '0722111112', 'emmagj@yahoo.com');
insert into INSTRUCTORI values(403, 'Patrascu', 'Mara', 'streetdance', '0722111113', 'mpatr@gmail.com');
insert into INSTRUCTORI values(404, 'Danca', 'Diana', 'dansuri africane', '0722111114', 'didiana@gmail.com');
```

Cursuri:

Am folosit TO_DATE pentru a asigura introducerea de date tip DATE în tabelă fără erori.

```
insert into CURSURI values(500, 400, 200, 'Hip Hop Basics', 'incepatori', '14:00', '15:00',
to_date('10-01-2021', 'dd-mm-yyyy'), to_date('20-09-2021', 'dd-mm-yyyy'), 'joi', 700);
insert into CURSURI values(501, 400, 200, 'Hip Hop Choreo', 'mediu', '15:00', '16:00',
to_date('10-01-2021', 'dd-mm-yyyy'), to_date('20-09-2021', 'dd-mm-yyyy'), 'joi', 800);
insert into CURSURI values(502, 402, 200, 'Popping', 'incepatori', '16:00', '17:00',
to_date('10-01-2021', 'dd-mm-yyyy'), to_date('20-09-2021', 'dd-mm-yyyy'), 'joi', 700);
insert into CURSURI values(503, 402, 200, 'Body Control', 'incepatori', '14:00', '15:00',
to_date('10-01-2021', 'dd-mm-yyyy'), to_date('20-09-2021', 'dd-mm-yyyy'), 'joi', 700);
insert into CURSURI values(504, 401, 201, 'Dans Contemporan', 'incepatori', '14:00', '15:00',
to_date('20-01-2021', 'dd-mm-yyyy'), to_date('30-09-2021', 'dd-mm-yyyy'), 'sambata', 700);
insert into CURSURI values(505, 401, 201, 'Contemporan Choreo', 'mediu', '15:00', '16:00',
to_date('20-01-2021', 'dd-mm-yyyy'), to_date('30-09-2021', 'dd-mm-yyyy'), 'sambata', 800);
insert into CURSURI values(506, 403, 201, 'Street Dance', 'incepatori', '17:00', '18:00',
to_date('20-01-2021', 'dd-mm-yyyy'), to_date('30-09-2021', 'dd-mm-yyyy'), 'sambata', 700);
insert into CURSURI values(507, 404, 202, 'Afro Basics', 'incepatori', '14:00', '15:00',
to_date('20-01-2021', 'dd-mm-yyyy'), to_date('30-09-2021', 'dd-mm-yyyy'), 'duminica', 700);
insert into CURSURI values(508, 404, 202, 'Dancehall', 'incepatori', '15:00', '16:00',
to_date('20-01-2021', 'dd-mm-yyyy'), to_date('30-09-2021', 'dd-mm-yyyy'), 'duminica', 700);
```

Înscrieri:

```
insert into INSCRIERI values(600, 100, 500, to_date('10-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(601, 104, 501, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(602, 101, 502, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(603, 100, 503, to_date('10-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(604, 102, 504, to_date('16-02-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(605, 102, 508, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(606, 103, 506, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(607, 105, 505, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(608, 106, 505, to_date('10-02-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(609, 107, 507, to_date('11-02-2021', 'dd-mm-yyyy'));
insert into INSCRIERI values(610, 104, 505, to_date('11-02-2021', 'dd-mm-yyyy'));
```

Plăți:

```
insert into PLATI values(600, 'cash', 700, to_date('10-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(601, 'cash', 800, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(602, 'card', 700, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(603, 'card', 700, to_date('10-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(604, 'cash', 700, to_date('16-02-2021', 'dd-mm-yyyy'));
insert into PLATI values(605, 'cash', 700, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(606, 'cash', 700, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(607, 'cash', 800, to_date('09-01-2021', 'dd-mm-yyyy'));
insert into PLATI values(608, 'cash', 800, to_date('10-02-2021', 'dd-mm-yyyy'));
insert into PLATI values(609, 'cash', 700, to_date('11-02-2021', 'dd-mm-yyyy'));
insert into PLATI values(610, 'cash', 800, to_date('11-02-2021', 'dd-mm-yyyy'));
```

Închirieri:

```
insert into INCHIRIERI values(800, 203, 'Manea', '0722555551', to_date('11-06-2021', 'dd-mm-yyyy'), '11:00', '12:00', 50);
insert into INCHIRIERI values(801, 203, 'Andrei', '0722555550', to_date('12-06-2021', 'dd-mm-yyyy'), '11:00', '12:00', 50);
insert into INCHIRIERI values(802, 201, 'Ionascu', '0722555551', to_date('11-06-2021', 'dd-mm-yyyy'), '09:00', '11:00', 120);
```

La final, am rulat comanda COMMIT pentru a salva înregistrările.