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.

1. **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.

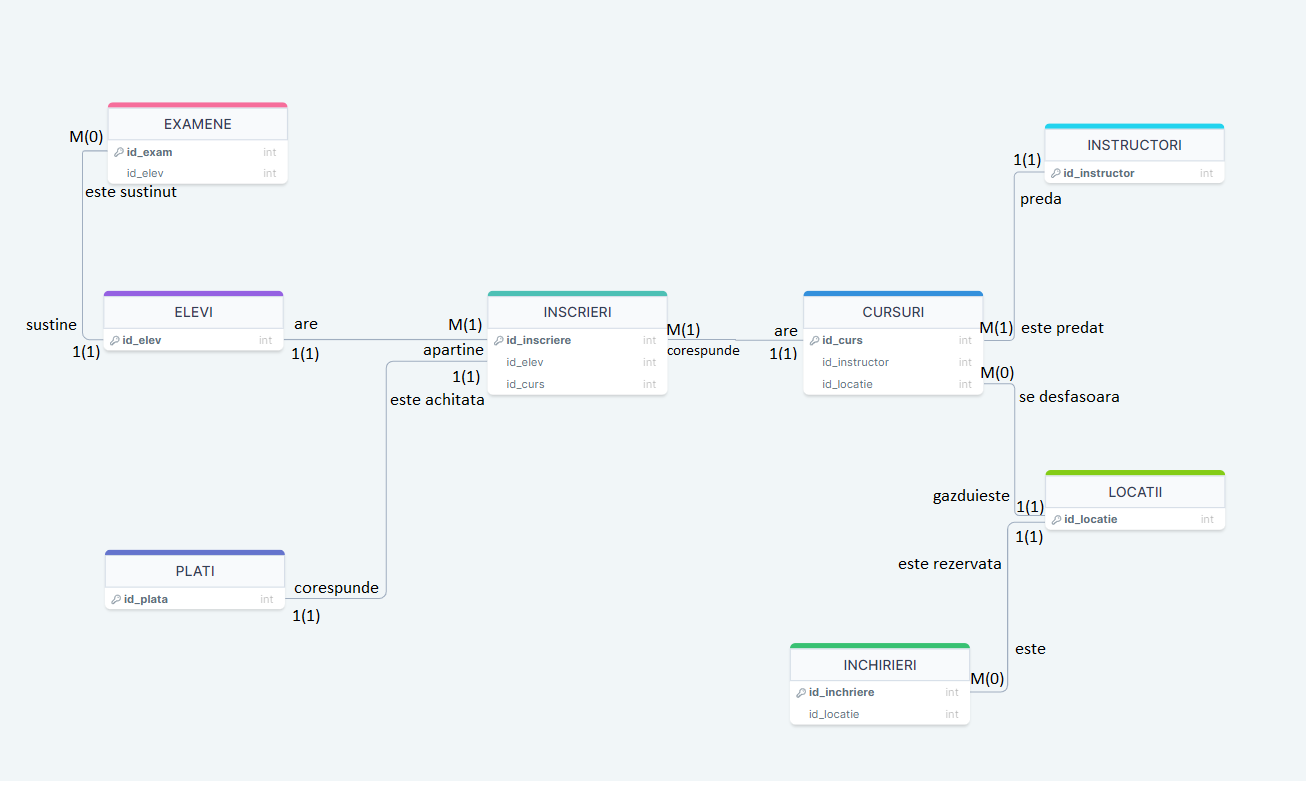
Instructori:

* + Un instructor poate preda mai multe cursuri.

Închirieri:

* + O închiriere se poate desfășura la o singură locație

1. **Diagrama entitate-relație:**

****

* 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 NUMBERși 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 propia 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 atribute 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; treuie 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 NUMBERși o cheie străină *id\_locatie,* cheia primară a tabelei Locații. Celelalte atribute 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\_închiriere:* 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 atributele 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 condiționeaă 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 desfășoara 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 atribute 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**

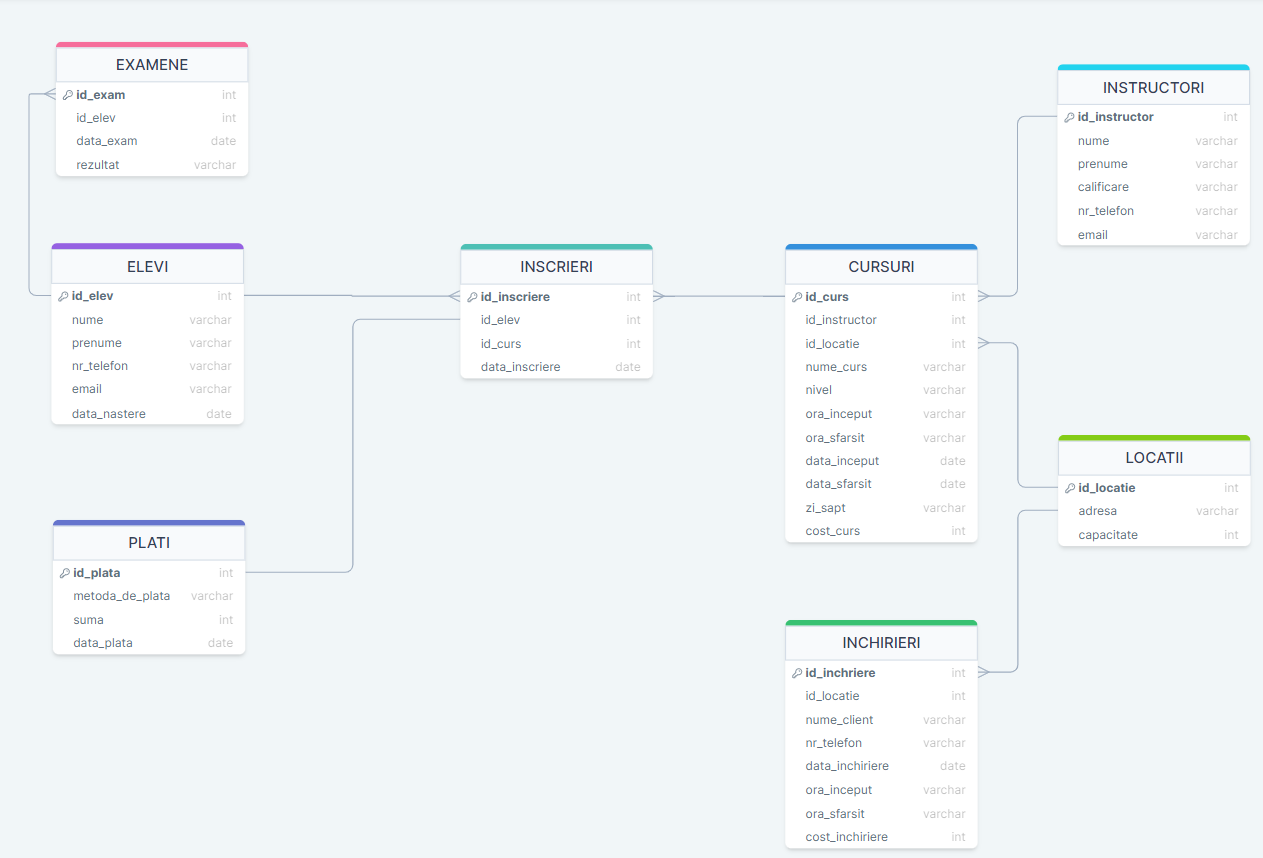
* + O plată trebuie să corespundă unei înscrieri.
  + O înscriere trebuie achitată printr-o plată.
  + O plată corespunde unei singure înscrieri.
  + O înscriere este achitată printr-o singură plată.

Locații – Inchirieri:

**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.

1. **Diagrama conceptuală:**



* 1. **Descrierea constrângerilor de integritate:**

Instructori:

* + Constrângere de tip PRIMARY KEY: ***pk\_instructori*** pentru atributul *id\_instructor.*
  + Constrângeri de tip NOT NULL: pentru atributele *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 atributele *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 atributele *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 preda.
  + Constrângere de tip CHECK: ***chk\_date\_curs*** pentru a compara atributele *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 atributele *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 atributele *nume, nr\_telefon,* deoarece aceste attribute sunt obligatorii.

Examene:

* + 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 atributele *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 atributele *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 atributele *metoda\_de\_plata, suma, data\_plata.*

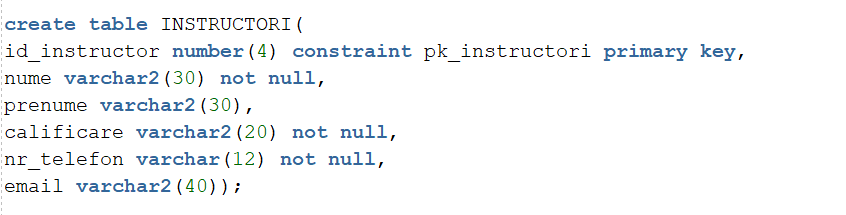
**4.2 Scheme relaționale:**

* 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);
* INSTRUCTORI(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\_inchriere#, 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).

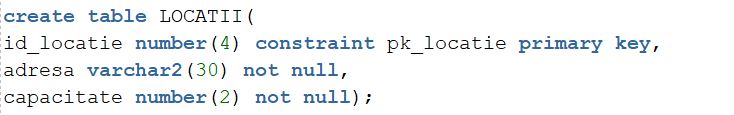
1. **Crearea tabelelor și inserarea de valori:**

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

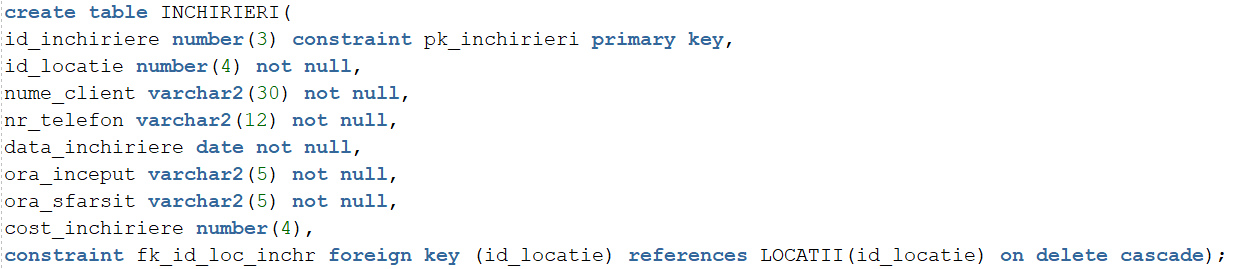
Instructori:



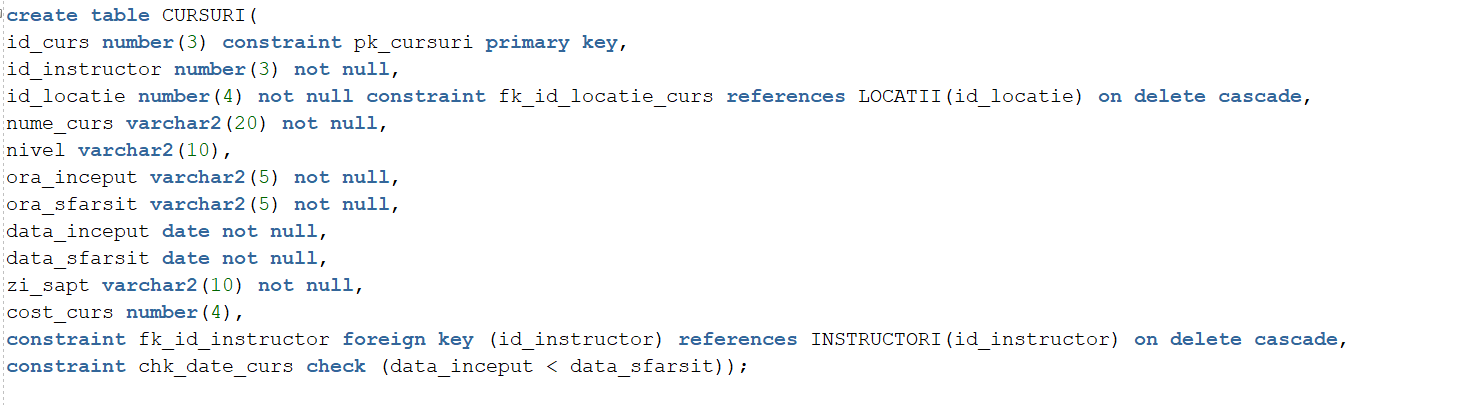
Locații:



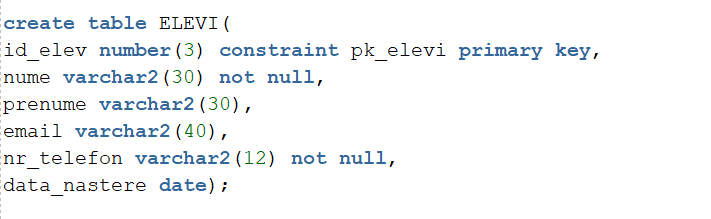
Închirieri:

****

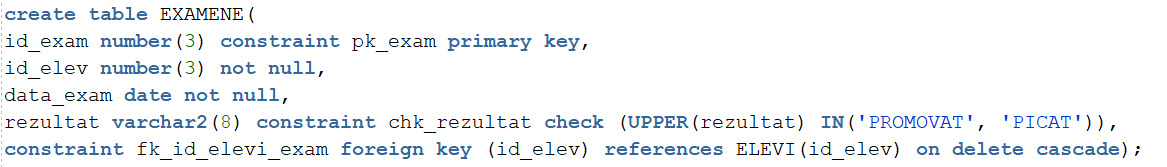
Cursuri:



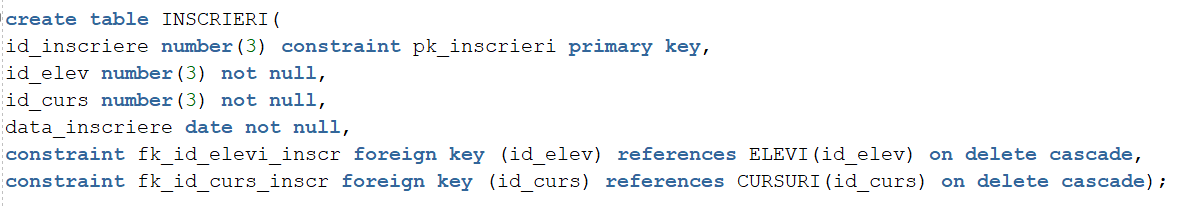
Elevi:



Examene:

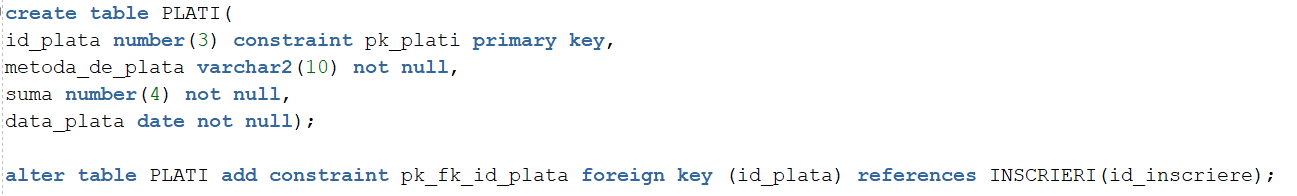


Înscrieri:



Plăți:

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

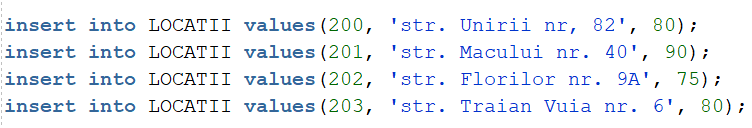


Pentru inserarea de valori, am folosit comanda INSERT INTO … VALUES.

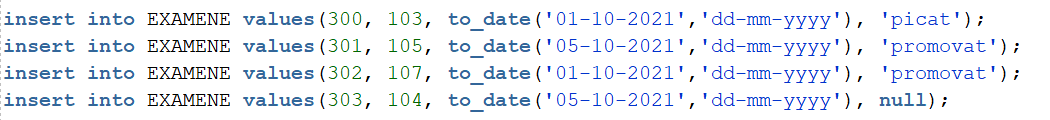
Elevi:



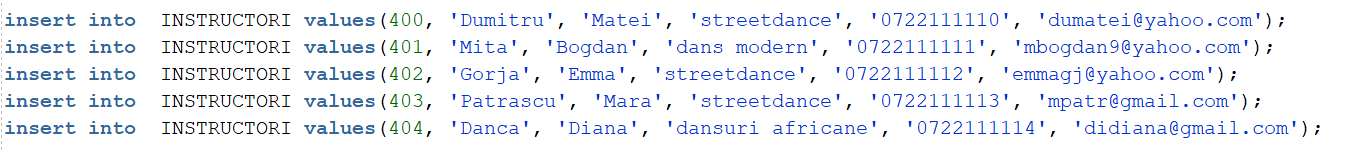
Locații:



Examene:



Instructori:

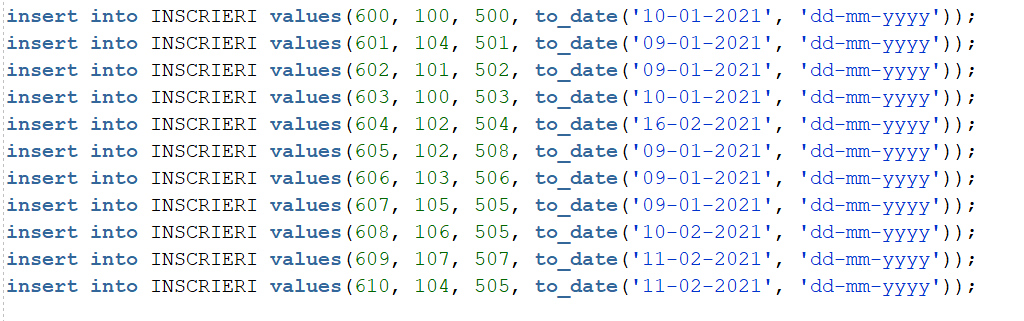


Cursuri:

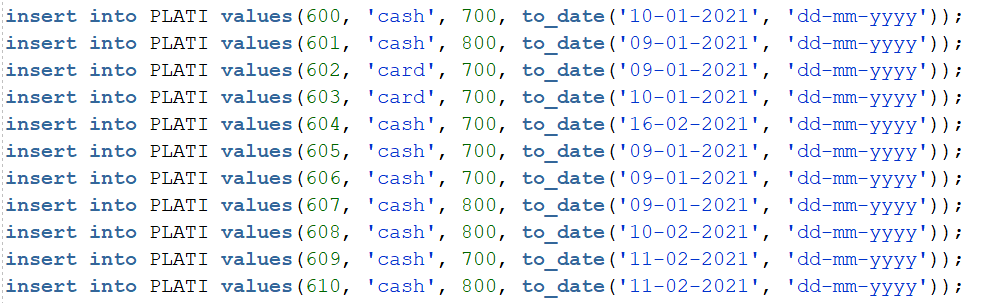
Am folosit TO\_DATE pentru a asigura introducerea de date tip DATE în tabelă fără erori.



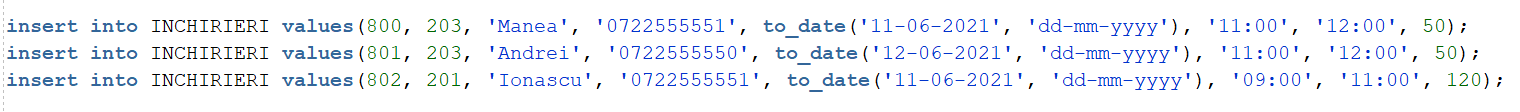
Înscrieri:



Plăți:



Închirieri:



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