

**Aplicație pentru evidența participării la cursuri online  
pentru o companie ce oferă servicii educaționale**

Colcer Alexia-Ioana

Grupa: 334 AA

## 1. Introducere

Baza de date ce urmează a fi implementată vine în ajutorul unei companii care ofera servicii educaționale, prin faptul că se ocupă de înregistrarea și stocarea informațiilor referitoare la participarea studenților și instructorilor la cursurile oferite de platforma online a acestei companii.

Principalele componente ale bazei de date vor fi:

- **Studenții** care se vor putea înscrie și participa la diferitele cursuri pe care le oferă aplicația.
- **Instructorii** calificați care vor ține cursurile pe platformă.
- **Cursurile** online la care vor putea lua parte studenții.
- **Lecțiile** sau modulele din cadrul cursurilor.
- **Examen final** ce va avea ca rezultat, în cazul completării acestuia, finalizarea cursului respectiv.

Deoarece, în mod evident, un student se poate înrola la mai multe cursuri în același timp și un curs cu siguranță poate avea înscriși mai mulți studenți, relația dintre aceste două tabele va fi una de tipul „many to many” și în consecință vom adauga tabela de legătură „Participări\_Cursuri”.

## 2. Definirea cerințelor

### 2.1. Lista funcționalităților

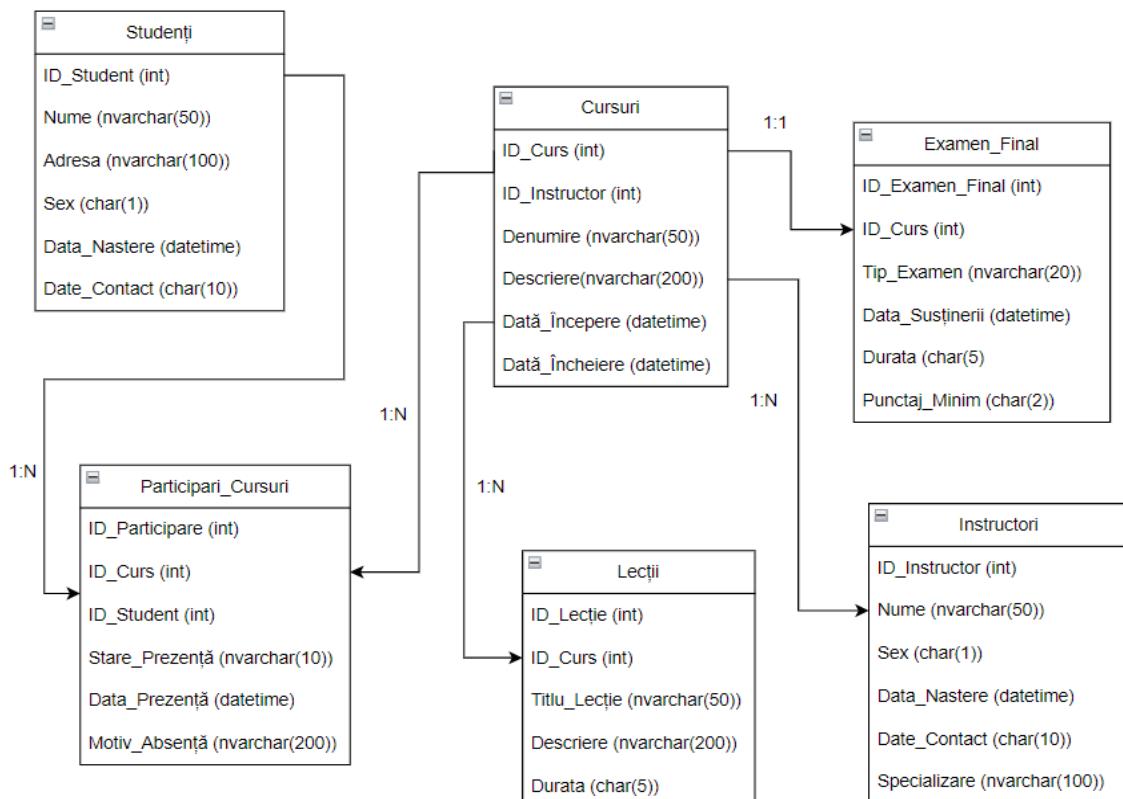
- Înregistrare Studenți: permite înregistrarea participanților la cursurile online, colectând informații precum nume, adresa, detalii de contact și altele.
- Gestionare Instructori: furnizează o platformă pentru gestionarea și actualizarea detaliilor, inclusiv experiența, calificările și programul de predare.
- Înscrieri la Cursuri: permite participantilor să se înscrie la cursurile dorite și furnizează confirmări automate de înscriere.
- Gestionare Examen Final: urmărește și administrează examenul final din cadrul unui curs, inclusiv data susținerii, tipul de examen (prezentare orală sau examen scris) și notarea acestuia.
- Lecții și Module: structurează lecțiile și modulele în cadrul cursurilor, furnizând acces ușor la conținutul educațional.
- Raportare Participare: generează rapoarte privind participarea la cursuri pentru fiecare participant, inclusiv numărul de cursuri la care au participat și progresul lor.

- Stare Cursuri: furnizează informații actualizate despre starea fiecărui curs (în desfășurare, încheiat, în pregătire).
- Autentificare și Securitate: asigură securitatea datelor și autentificarea participantilor, instructorilor și administratorilor.

## 2.2. Identificarea componentelor

- Front-end: o interfață web care oferă o experiență de navigare ușoară și interacțiune fluidă pentru participanți și instructori în cadrul acestei aplicații de gestionare a cursurilor online.
- Back-end: va gestiona funcționalitățile precum înregistrarea și gestionarea cursurilor, examenelor finale, lecțiilor. De asemenea, baza de date stochează informații cheie despre participanți, instructori și cursuri.
- Sistem de Autentificare: un sistem de autentificare sigur care protejează datele și asigură accesul controlat pentru participanți, profesori și administratori, oferind confidențialitate și securitate în utilizarea aplicației.
- Gestionare Examen\_Final: o componentă pentru programarea, gestionarea și evaluarea examenelor finale necesare finalizării cursurilor.
- Organizare Cursuri: un sistem de organizare a cursurilor, modulelor sau lecțiilor, cu un motor de căutare care ajută participanții să găsească rapid conținutul dorit.

## 3. Proiectarea bazei de date



Dupa cum se poate observa și din diagrama de mai sus, în baza de date vom avea urmatoarele tabele:

- Studenți
- Cursuri
- Instructori
- Lecții
- Examen\_Final
- Participari\_Cursuri

## Constrângeri de integritate

Voi face urmatoarea considerație:

- Fiecare curs va fi predat/ținut de 1 singur instructor, însă un instructor poate să poată să predice mai multe cursuri diferite.

### 1. Constrângeri de tip CHECK

Examenul final din cadrul fiecărui curs va putea fi doar de 2 tipuri (se regăsește în tabela Examen-Final):

- Examen teoretic
- Examen practic

Starea prezenței (din tabela Participări\_Cursuri) poate lua doar următoarele valori:

- Prezent
- Absent
- Întârziat
- Motivat

Sex-ul unei persoane (din tabelele Studenți și Instructori) poate lua doar 2 valori:

- F
- M

2. Constrângeri de tip UNIQUE:

- Date\_Contact (din tabelele Studenți si Instructori)
- Denumire (din tabela Cursuri)

3. Constrângeri de tip Primary Key:

- ID\_Student (din tabela Studenți)
- ID\_Curs (din tabela Cursuri)
- ID\_Instructor (din tabela Instructori)
- ID\_Examen\_Final (din tabela Examen\_Final)
- ID\_Lecție (din tabela Lecții)
- ID\_Participare (din tabela Participări\_Cursuri)

4. Constrângeri de tip Foreign Key:

- ID\_Curs (din tabela Examen\_Final cu referire la ID\_Curs din tabela Cursuri)
- ID\_Curs (din tabela Lecții cu referire la ID\_Curs din tabela Cursuri)
- ID\_Student și ID\_Curs (din tabela Participări\_Cursuri cu referire la ID\_Student din tabela Studenți si ID\_Curs din tabela Cursuri)
- ID\_Instructor (din tabela Cursuri cu referire la tabela Instructori)

5. Constrângeri de tip NULL (atribute care au voie sa ia valoarea NULL):

- Adresa, Sex, Data\_Naștere (din tabela Studenți, pentru că nu sunt informații foarte relevante)
- Sex, Data\_Naștere (din tabela Instructori)
- Descriere (din tabelele Cursuri si Lecții)
- Motiv\_Absență (din tabela Participări\_Cursuri, pentru că in cazul in care studentul nu are motivare pentru absență sau este prezent nu se va completa)

## Functionarea Aplicatiei :

Aplicatia web pe care ai dezvoltat-o ofera un mediu accesibil si eficient pentru a gestiona informatii legate de cursuri si studenti intr-un context educational. Utilizatorii pot naviga usor prin diferite functionalitati ale aplicatiei, inclusiv cautarea rapida a lecțiilor in functie de ID-ul cursului si adaugarea sau actualizarea informatiilor despre studenti, cum ar fi numele, adresa si datele de contact. De asemenea, aplicatia permite utilizatorilor sa gestioneze informatii legate de prezentarea studentilor la cursuri, inregistrand starea prezentelor si motivarea absentei. Prin intermediul interfetei web prietenoase, utilizatorii pot accesa informatii detaliate despre studenti si cursuri, contribuind la o administrare mai eficiente si transparenta a datelor in context educational. Toate acestea sunt realizate prin conectarea la o baza de date SQL Server, asigurand integritatea si securitatea datelor.

## Interogari Simple :

```
"SELECT S.ID_Student, S.Nume, S.Adresa, S.Sex, S.Data_Nastere, S.Date_Contact FROM dbo.STUDENTI AS S")  
  
'''SELECT I.Nume , C.Denumire, C.Descriere, C.Data_Incepere, C.Data_Incheiere  
      FROM dbo.INSTRUCTORI AS I INNER JOIN CURSURI AS C ON C.ID_Instructor = I.ID_Instructor'''  
  
'''SELECT S.Nume , C.Denumire, PC.Stare_Prezenta, PC.Data_Prezenta, PC.Motiv_Absenta  
      FROM dbo.PARTICIPARE_CURSURI AS PC INNER JOIN CURSURI AS C ON C.ID_Curs = PC.ID_Curs  
          INNER JOIN STUDENTI AS S ON S.ID_Student= PC.ID_Student'''  
  
cursor.execute('''SELECT C.Denumire , L.Titlu_Lectie, L.Descriere, L.Durata  
      FROM dbo.CURSURI AS C JOIN LECTII AS L ON C.ID_Curs = L.ID_Curs  
      WHERE C.Denumire = ? ''', ncurs).fetchall()
```

## Interogari Complexe :

```
''' SELECT I.Nume, C.Denumire, EM.Tip_Examen, EM.Data_Sustinerii, EM.Durata, EM.Punctaj_Minim,  
      (SELECT COUNT(*)  
        FROM PARTICIPARE_CURSURI AS PC  
        WHERE PC.ID_Curs = C.ID_Curs) AS Total_Studenti  
      FROM CURSURI AS C  
      JOIN INSTRUCTORI AS I ON C.ID_Instructor = I.ID_Instructor  
      JOIN EXAMEN_FINAL AS EM ON EM.ID_Curs = C.ID_Curs  
      WHERE I.Nume = ?  
      ORDER BY EM.Punctaj_Minim DESC ''', numeprof).fetchall()
```

```

''' SELECT I.Nume, C.Denumire, EM.Tip_Examen, EM.Data_Sustinerii, EM.Durata, EM.Punctaj_Minim,
        (SELECT COUNT(*)
         FROM PARTICIPARE_CURSURI AS PC
         WHERE PC.ID_Curs = C.ID_Curs) AS Total_Studenti
        FROM CURSURI AS C
        JOIN INSTRUCTORI AS I ON C.ID_Instructor = I.ID_Instructor
        JOIN EXAMEN_FINAL AS EM ON EM.ID_Curs = C.ID_Curs
        WHERE C.Denumire = ? AND EM.Tip_Examen = ? ''' , numecurs, tipexam).fetchall()

```

## INSERT:

```

INSERT INTO dbo.STUDENTI ( Nume, Adresa, Sex, Data_Nastere , Date_Contact) "
VALUES ( ?, ?, ?, ? , ?)" , numestud, adresa, sex, datanastere, datecontact)

```

```

''' INSERT INTO dbo.PARTICIPARE_CURSURI (ID_Curs, ID_Student, Stare_Prezenta, Data_Prezenta, Motiv_Absenta)
        SELECT C.ID_Curs, S.ID_Student, ?, ?, ?
        FROM CURSURI C, STUDENTI S
        WHERE C.Denumire = ? AND S.Nume = ?''' , numecurs, numestud, stareprezenta, dataprezenta, motivare)

```

## UPDATE:

```

'UPDATE dbo.STUDENTI SET Adresa = ? , Data_Nastere = ?, Date_Contact = ?
WHERE ID_Student = (SELECT DISTINCT S.ID_Student FROM STUDENTI S "
WHERE S.Nume = ?)" , adresa, datanastere, datecontact, numestud)

```

```

UPDATE dbo.PARTICIPARE_CURSURI "
SET Stare_Prezenta = ? , Data_Prezenta = ? , Motivare = ? "
WHERE ID_Student = (SELECT DISTINCT S.ID_Student FROM STUDENTI S WHERE S.Nume = ?) "
AND ID_Curs = (SELECT C.ID_Curs FROM CURSURI C WHERE C.Denumire = ?)" , stareprezenta, dataprezenta, motivare, numestud, numecurs)

```

## DELETE:

```

id_students_to_delete = cursor.execute('SELECT S.ID_Student FROM dbo.STUDENTI S WHERE S.Nume = ?' , student_name).fetchone()

'DELETE FROM dbo.PARTICIPARE_CURSURI WHERE ID_Student = ?' , id_students_to_delete)

'DELETE FROM dbo.STUDENTI WHERE ID_Student = ?' , id_students_to_delete)

```