

Tema proiectului: Gestiunea unei orchestre

Cadru didactic coordinator:

Tizu Matei-Victor

Student:

Radion Ciprian, 1309B

Descrierea proiectului:

O orchestra are nevoie de ajutor pentru a putea organiza mai multe concerte. Fiecare concert are o anumita data. Un concert are un singur dirijor, iar un dirijor dirijeaza un singur concert. Un concert poate avea mai multe piese în repertoriu, iar o piesă poate apărea în mai multe concerte. Un concert poate implica mai mulți muzicieni, iar un muzician poate participa la mai multe concerte. Un muzician cântă la un singur instrument, iar un instrument poate fi asociat cu mai mulți muzicieni.

Tehnologii folosite FRONT-END:

- **HTML:** este un limbaj standardizat utilizat pentru a crea structura de bază a paginilor web. Acesta constă într-un set de elemente sau etichete, fiecare având un rol specific în organizarea și afișarea conținutului pe pagină.
- **Bootstrap:** este un framework front-end care simplifică dezvoltarea paginilor web, permițând dezvoltatorilor să creeze rapid și eficient interfețe responsive și estetice. Bootstrap furnizează o colecție extinsă de componente, stiluri predefinite și funcționalități JavaScript, facilitând astfel procesul de construire a paginilor web moderne.

BACK-END:

- **FLASK:** este un framework web open-source care se concentrează pe simplitate și extensibilitate. Flask oferă un set de instrumente și facilități esențiale pentru construirea aplicațiilor web eficiente și ușor de întreținut.
- **CLICK:** este un pachet Python folosit pentru a crea interfețe în linia de comanda, într-un mod ușor de înțeles, eficient și ușor de utilizat, cu cât mai puțin cod posibil.
- **WERKZEUG:** WERKZEUG ESTE O BIBLIOTECĂ UTILITARĂ PENTRU LIMBAJUL DE PROGRAMARE PYTHON , CU ALTE CUVINTE UN SET DE INSTRUMENTE PENTRU APLICAȚII WSGI (WEB SERVER GATEWAY INTERFACE) ȘI ESTE LICENȚIAT SUB LICENȚĂ BSD . WERKZEUG POATE REALIZA OBIECTE SOFTWARE PENTRU FUNCȚII DE SOLICITARE, RĂSPUNS ȘI FUNCȚII UTILITARE.
- **JINJA2:** este un motor de șabloane pentru limbajul de programare Python și este frecvent folosit în aplicațiile web bazate pe framework-ul Flask. A fost dezvoltat pentru a permite integrarea eficientă a șabloanelor în aplicațiile Python, permițând în același timp o sintaxă ușor de înțeles și de utilizat.
- **CX_ORACLE:** este un modul Python care furnizează o interfață pentru accesul la bazele de date Oracle. Este un driver oficial Oracle pentru Python și permite dezvoltatorilor să conecteze aplicațiile Python la baze de date Oracle, să execute interogări SQL și să manipuleze datele într-un mod eficient

În urma analizei cerintelor rezulta:

1.Entitatea concerte:

- id concert: cheie primara pentru identificarea unica a fiecarui concert
- nume concert: nu poate fi nul si are un constraint pentru a asigura ca are o lungime mai mare de 1 caracter
- data
- oras: are un constraint pentru a asigura ca are o lungime mai mare de 1 caracter

2.Entitatea dirijori:

- id dirijor: cheie primara
- nume: nu poate fi nul si are un constraint pentru a asigura ca are o lungime mai mare de 1 caracter
- email: contine un constraint pentru adresa de e-email
- numar de telefon

3.Entitatea muzicieni:

- id musician: cheie primara
- nume: constraint pentru o lungime mai mare de 1 si nu poate fi nul
- email: constraint pentru format valid
- numar de telefon

4.Entitatea piese:

- numele autorului piesei: constraint pentru caracter si not null
- titlul: constraint pentru caracter si not null
- id piesa: cheie primara

5.Entitatea instrumente:

- id instrument :cheie primara
- categorie: costringere pentru
categorie('coarda_arcus','complexe','percutie','suflat')
- nume: costringere not null sim ai mare de 1 caracter

Si tabelele rezultate din relatiile M:M :

6. Participari:

- id musician: cheie straina

-id concert: cheie straina

-loc pe scena

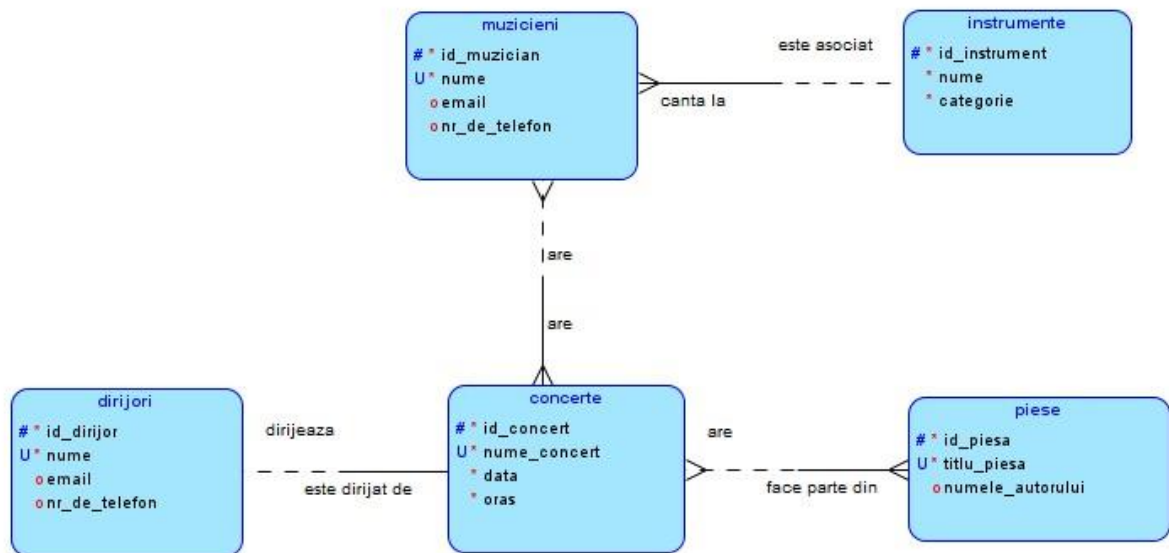
7. Repertoriu:

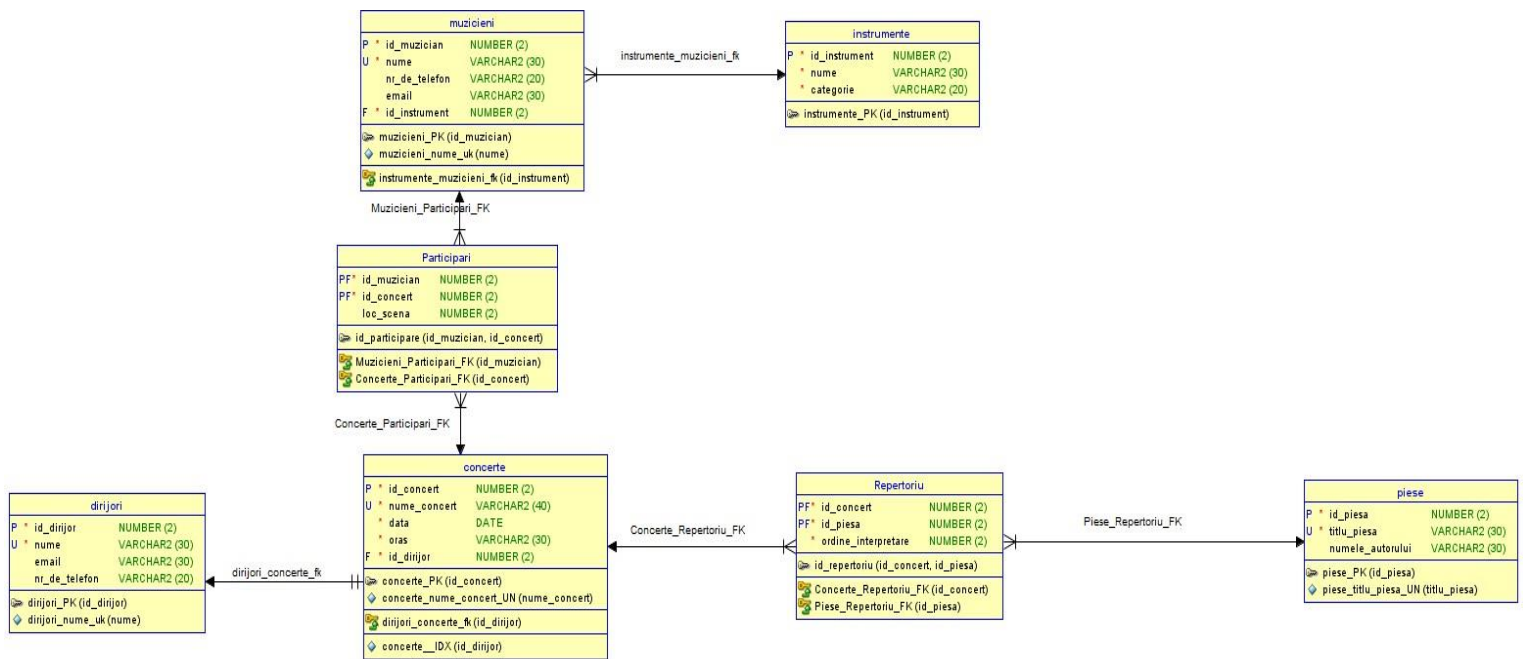
-id concert: cheie straina

-id piesa: cheie straina

-ordine interpretare piesa

Structura si inter-relationarea tabelelor:





Inter-relatiile sunt create prin chei straine, asigurand astfel coerenta si integritatea datelor in intreaga baza de date. De asemenea, exista constrangeri pentru a se asigura ca datele introduse respecta anumite reguli si formate. Aceasta este o schema bine structurata care incurajeaza normalizarea datelor si faciliteaza gestionarea lor într-un mod coerent si eficient.

Constrangerile de tip length adaugat au fost necesare pentru a asigura ca fiecare varchar contine un nume/data valida si pentru a preveni introducerea de date incorecte sau goale.

Cele de tip unique pentru a preveni repetarea unui nume, unde a fost necesar(in tabela concert(num_e_concert), tabela dirijori(num_e),tabela muzicieni(num_e)).

Constrangerea de la instrumente(categorie_ck) asigura ca categoria instrumentului este una dintre cele valide specificate, contribuind la coerenta si consistenta datelor.

Constrangerea pentru email(din tabela dirijori si tabela muzicieni) contribuie la integritatea datelor, verificand daca adresa de email are un format valid.

Interfata aplicatiei

1.Functia Select:

```
@app.route('/participari')
def participari():
    participari=[]

    cur=con.cursor()
    cur.execute('select * from participari')
    for result in cur:
        participare={}
        participare['id_muzician']=result[0]
        participare['id_concert']=result[1]
        participare['loc_scena']=result[2]
        participari.append(participare)
```

participari

Adauga participare

id muzician	id concert	loc scena
1	1	1

2.Functia Delete

```
@app.route('/del_dirijor', methods=['GET', 'POST'])
def del_dir():
    id_dirijor=request.form['id_dirijor']
    cur=con.cursor()
    cur.execute('DELETE FROM repertoriu WHERE id_concert IN (SELECT id_concert FROM concerte WHERE id_dirijor = :id)', {'id': id_dirijor})
    cur.execute('DELETE FROM participari WHERE id_concert IN (SELECT id_concert FROM concerte WHERE id_dirijor = :id)', {'id': id_dirijor})
    cur.execute('DELETE FROM concerte WHERE id_dirijor = :id', {'id': id_dirijor})
    cur.execute('DELETE FROM dirijori WHERE id_dirijor = :id', {'id': id_dirijor})
    cur.execute('commit')
    return redirect('/dirijori')
```

3.Functia Update

```
nr_de_telefon=request.form['nr_de_telefon']
#print(name)
cur=con.cursor()
cur.execute("UPDATE dirijori SET nume=: name,email=:email,nr_de_telefon=:nr_de_telefon WHERE id_dirijor=:dir",name=name,email=email,nr_de_telefon=nr_de_telefon)
con.commit()
return redirect(url_for('dir'))
```

dirijori

Adauga dirijor

ID	nume	email	numar de telefon	Editare/Stergere
2	Husi	husi@gmail.com	0317756	<div>Editeaza dirijor</div> <div>Sterge dirijor</div>
1	Radion Ciprian	radionciprian@gmail.com	0712934568	<div>Editeaza dirijor</div> <div>Sterge dirijor</div>

4.Functia Insert

```
fields=['id_concert','nume_concert','data','oras','id_dirijor']
print(fields)
cur.execute("INSERT INTO CONCERTE (id_concert,nume_concert,data,oras,id_dirijor) VALUES (:1,:2,TO_DATE(:3, 'YYYY-MM-DD'),:4,:5)",(conc,request.form['nume_conce
```