

# Baze de date – proiect

Dumitriu Denis-Gabriel , grupa 1311A

Cadru didactic coordonator - Mironeanu Catalin

## Cuprinsul documentatiei:

Pagina 1 : Descrierea proiectului / scopul aplicatiei

Pagina 2: Model Logic + structura si inter-relationarea  
tabelelor

Pagina 3: Model Relational + pasii normalizarii

Descriere Coloane+Tip+Constrangeri la tabelele

Pagina 4: Cinema si Sala

Pagina 5: Client si Profil Client

Pagina 6: Film si Proiectie

Pagina 7: Rezervare si Feedback

## Descrierea proiectului / Scopul aplicatiei

Scopul proiectului este dezvoltarea unui sistem de gestionare pentru un lant de cinematografe cu numeroase sali, in care au loc zilnic proiectii de filme, si care permite utilizatorilor sa vizualizeze cinematografele din apropiere, recenziile din experienta altor utilizatori la cinema, cele mai noi filme in toate formatele disponibile, orele si zilele de proiectie, sa isi creeze un profil si sa efectueze rezervari fie prin achitarea biletului online, fie la intrarea in sala, inainte de inceperea filmului.

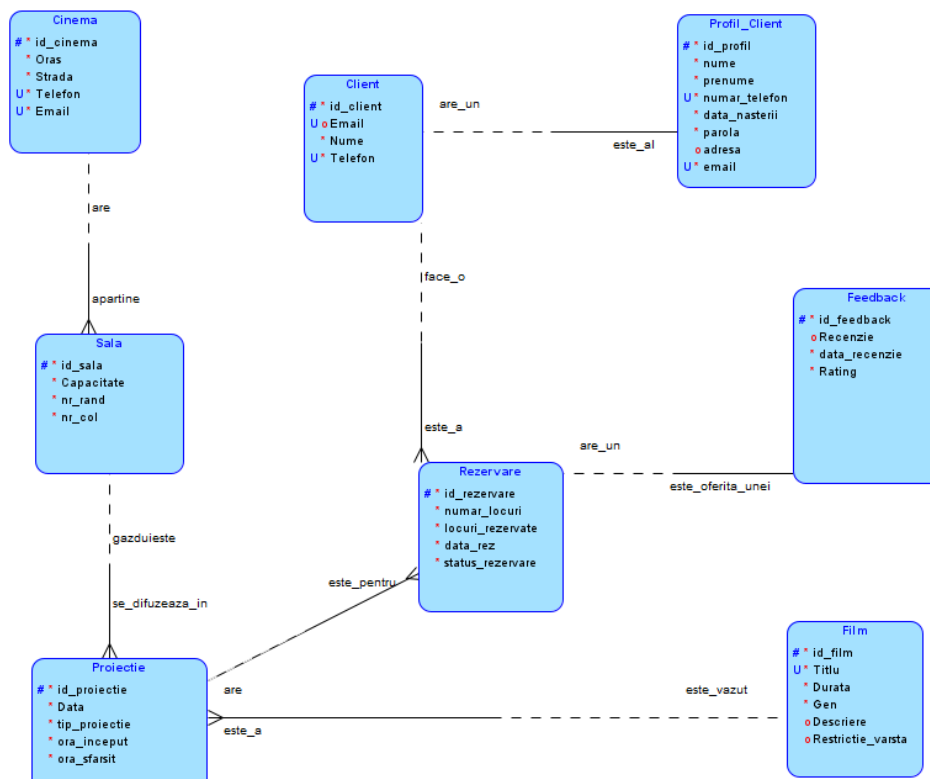
Structura bazei de date este organizata prin tabelele aferente, continand toate datele necesare pentru o functionare simpla si sigura, care permite si loc de imbunatatiri si noi attribute sau tabele (de exemplu: o tabela pentru barul de snacks-uri si bauturi, in care clientul are la dispozitie un meniu cu suc, popcorn, nachos sau dulciuri).

Ca administrator al bazei de date, pot vizualiza popularitatea fiecarui cinematograf in functie de cat de frecventat este de catre clienti zilnic, pot actualiza lista cu proiectiile filmelor in momentul in care apare un film nou sau unul ruleaza de prea mult timp, pot modifica detalii despre cinematografe (cum ar fi numarul salilor atunci cand se deschide una noua sau una este in renovare) si pot consulta rezervarile realizate de clienti, inclusiv locurile deja ocupate.

Inspiratia proiectului a venit din franciza Cinema City si modul in care aceasta gestioneaza experienta utilizatorului pe site. Sistemul ofera informatii utile clientilor si faciliteaza realizarea rezervarilor pe baza datei calendaristice si a unui interval orar, permitand vizualizarea locurilor pe care doresc sa le rezerve, folosind o interfata grafica.

O motivatie a alegerii acestui proiect este faptul ca am realizat un proiect individual, constand intr-o aplicatie ce poate fi utilizata din perspectiva unui client, dar si din cea a administratorului (server). Din perspectiva clientului, aplicatia permite, prin intermediul unei interfete grafice, alegerea zilei si orei dorite pentru vizionarea unui film, vizualizarea locurilor ocupate si disponibile, iar in urma efectuarii unei rezervari, pe baza unui email, stochez detaliile fiecarei rezervari intr-o baza de date cu informatiile utile despre rezervarea clientului, acesta primind pe email o confirmare si avand posibilitatea de a anula rezervarea. Ca administrator, pot observa in log-uri actualizarile tabelei si pot modifica orice detaliu al informatiilor din interfata grafica.

Proiectul are in vedere corectitudinea datelor inserate in tabele, fiecare tabel avand verificari care impiedica introducerea unor valori eronate. Structura bazei de date este construita astfel incat tabelele sa fie legate intre ele in mod logic, prin chei primare si straine.



Structura si inter-relationarea tabelor se pot observa in imaginea de mai sus. Sunt prezente 8 tabele, fiecare avand atribute diverse, care vor fi enumerate si descrise detaliat, inclusiv dimensiunile si constrangerile fiecaruia.

In proiect sunt prezente doar relatii de tip 1:1 sau 1:N, tabela aflata in dreapta primind ca Foreign Key id-ul (Primary Key-ul) tabelii din stanga.

### Relatiile de 1:N din proiect sunt intre tabelele:

Cinema->Sala (Un cinema are mai multe sali)

Sala->Proiectie (O sala gazduieste zilnic mai multe proiectii)

Film->Proiectie (Un film poate fi vazut in mai multe proiectii diferite)

Proiectie->Rezervare (O proiectie are mai multe rezervari simultan)

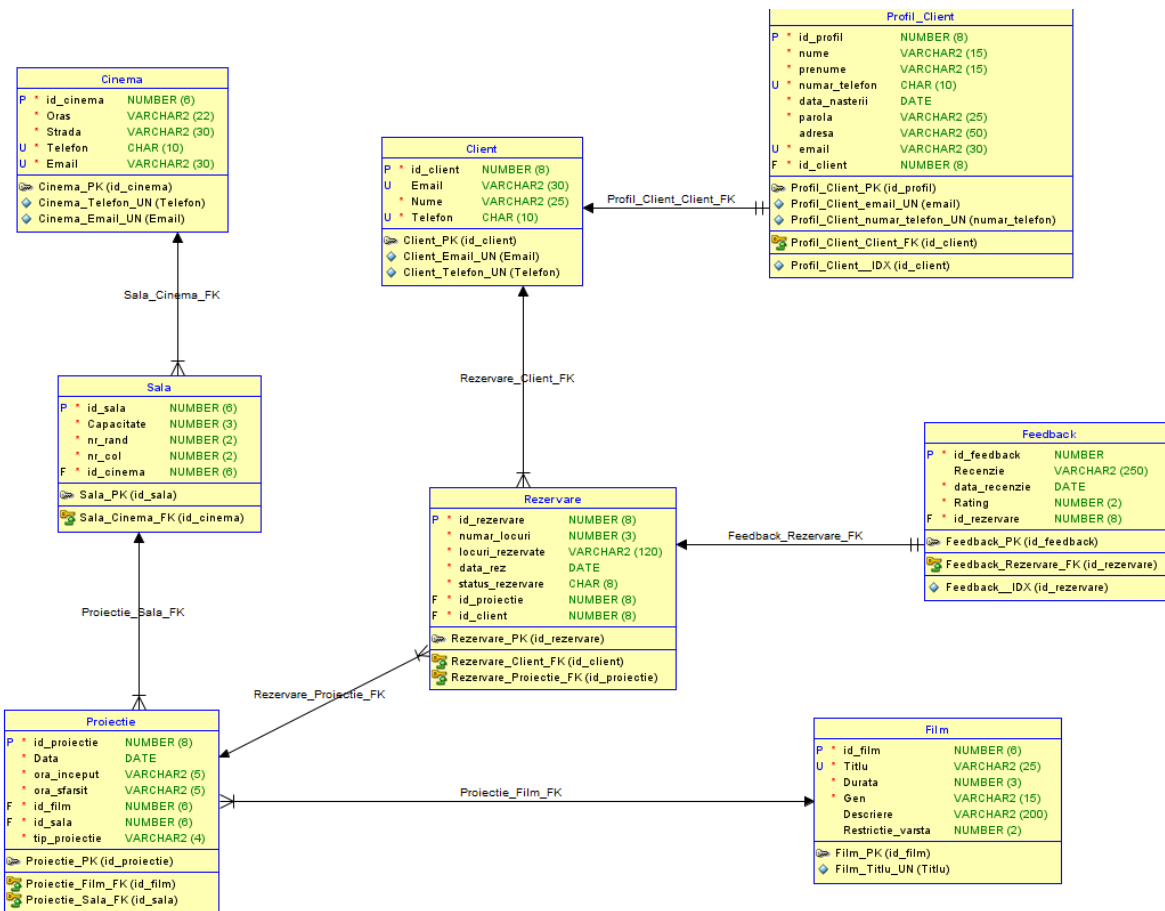
Client->Rezervare (Un client poate face mai multe rezervari)

### Relatiile 1:1 sunt urmatoarele:

Rezervare-Feedback (O rezervare are un singur feedback)

Client-Profil\_Client (un client are un singur profil)

Proiectul respecta legile normalizarii, a.i. organizarea nu permite sa existe informatii duplicate (nu pot avea 2 clienti cu acelasi email, 2 filme cu acelasi titlu, iar de asemenea fiecare tabela contine un ID ca si cheie primara, asigurand lipsa dublurilor), informatiile sunt stocate in tabela potrivita, relatiile sunt logice, dupa cum am discutat anterior.



Prima lege a normalizarii spune ca nu trebuie sa existe coloane repetitive, sau grupuri de attribute care sa se repete, fiecare camp continand o singura valoare. (in exemplu, clientul poate introduce un singur numar de telefon, sau un film poate avea un singur gen, nu o combinatie a mai multe genuri, nota din feedback oferita unei rezervari este un singur numar de la 1 la 10, un cinema poate avea un singur email, sau o singura adresa)

A doua lege impune ca toate coloanele sa depinda doar de cheia primara si sa respecte deja prima lege. Daca am fi avut o cheie primara compusa, atunci nu am fi fost in 2NF (decat daca attributele depindeau de intreaga cheie primara compusa), deoarece unele attribute ar fi depins doar de una din cele 2 parti ale cheii primare. Dar fiecare tabela avand o singura cheie primara, id-ul, este respectata a doua lege a normalizarii.

Si pentru a respecta legea a 3-a trebuie sa fie deja in a doua forma si ca fiecare coloana non-cheie sa nu depinda deloc de nicio alta coloana non-cheie. In cazul tabelor si atributelor din acestea, nu exista nicio astfel de situatie. Un contra-exemplu ar fi fost daca in tabela Proiectie aveam si attributele din Film, dar fiindca attributele nu depind de Proiectie, ci de Film, s-ar fi creat o dependenta tranzitiva care nu ar mai fi respectat a 3-a lege a normalizarii (durata nu ar fi depins de id\_proiectie, ci de id\_film sau Titlu)

Astfel, sunt respectate legile normalizarii in proiect pentru fiecare tabela si fiecare atribut

## Tabelul Cinema

- id\_cinema (NUMBER(6)): identificator unic pentru fiecare cinema; tip numeric pentru a putea folosi secventa de auto-increment. (PK)
- Oras (VARCHAR2(22)): denumirea orasului; lungimea 22 este suficienta chiar si pentru orasul cu cel mai lung nume: **Drobeta-Turnu Severin** + constrangere check ca lungimea sa fie mai mare ca 1
- Strada (VARCHAR2(30)): adresa stradala; lungimea aleasa pentru a include strazi lungi. + check ca lungimea este mai mare ca 1
- Telefon (CHAR(10)): numar fix de telefon mobil format din 10 cifre; CHAR pentru lungime fixa si verificare cu expresie regulata. + constrangere sa aiba formatul unui numar de telefon din RO, sa inceapa cu 07 si poate contine cifre intre 0 si 9
- Email (VARCHAR2(30)): adresa de email; lungimea de 30 caractere este suficienta pentru majoritatea adreselor. + constrangere sa aiba tipul unui email, format cu '@' si '.'

## Tabelul Sala

- id\_sala (NUMBER(6)): identificator unic sala ce contine autoincrement. (PK)
- Capacitate (NUMBER(3)): numarul total de locuri; + constrangere de tip list of ranges cu valori intre 50 si 300. (default 100)
- nr\_rand, nr\_col (NUMBER(2)): numarul de randuri si coloane; + constrangere ca val >4. (default 10)
- id\_cinema (NUMBER(6)): referinta la cinema. (FK)

## **Tabelul Client**

- id\_client (NUMBER(8)): identificator unic pentru client; NUMBER pentru secventa auto-increment de asemenea. (PK)
- Email (VARCHAR2(30)): optional, dar unic; lungimea de 30 este standard pentru adrese de email uzuale. + constrangere sa aiba tipul unui email, format cu '@' si '.'
- Nume (VARCHAR2(25)): nume client; 25 caractere suficient pentru persoane care au chiar si 2 prenume. + constrangere check ca lungimea sa fie mai mare ca 1
- Telefon (CHAR(10)): numar de telefon mobil; CHAR pentru lungime fixa. + constrangere sa aiba formatul unui numar de telefon din RO, sa inceapa cu 07 si poate contine cifre intre 0 si 9

## **Tabelul Profil\_Client**

- id\_profil (NUMBER(8)): identificator unic al profilului cu autoincrement. (PK)
- nume, prenume (VARCHAR2(15)): lungime aleasa pentru nume obisnuite. + constrangere check ca lungimea sa fie mai mare ca 2
- numar\_telefon (CHAR(10)): telefon profil, fix la 10 caractere. + constrangere sa aiba formatul unui numar de telefon din RO, sa inceapa cu 07 si poate continue cifre intre 0 si 9
- data\_nasterii (DATE): data nasterii clientului; tip DATE pentru operatii ulterioare de varsta. + trigger ca data nasterii sa nu fie in viitor
- parola (VARCHAR2(25)): parola utilizator; lungime suficienta pentru securitate. + constrangere check ca lungimea sa fie mai mare ca 8
- adresa (VARCHAR2(50)): adresa completa, optionala; lungime mai mare pentru detalii aditionale.
- email (VARCHAR2(30)): adresa email asociata profilului. + constrangere sa aiba tipul unui email, format cu '@' si '.'
- id\_client (NUMBER(8)): referinta la client; tip numeric compatibil cu id\_client. (FK)

## **Tabelul Film**

- id\_film (NUMBER(6)): identificator unic film ce are autoincrement. (PK)
- Titlu (VARCHAR2(25)): titlu film; lungime 25 pentru titluri scurte. + constrangere check ca lungimea sa fie mai mare ca 1
- Durata (NUMBER(3)): durata in minute; + constrangere list of ranges cu valori intre 60 si 240. – valoare exprimata in minute (default 90)
- Gen (VARCHAR2(15)): genul filmului; lungime suficienta pentru genuri standard. + constrangere cu list of ranges in care am pus cele mai comune genuri de filmele
- Descriere (VARCHAR2(200)): descriere optionala; lungime mai mare pentru detalii. + check ca poate sa lipseasca descrierea sau ca lungimea sa fie mai mare ca 10
- Restrictie\_varsta (NUMBER(2)): (optional) varsta minima recomandata; valori mici, tip numeric. + constrangere list of values cu valorile: 3, 7, 12, 15 si 18 ani

## **Tabelul Proiectie**

- id\_proiectie (NUMBER(8)): identificator unic proiectie si are autoincrement. (PK)
- Data (DATE): ziua proiectiei. + trigger ca ziua proiectiei sa fie in viitor, nu in trecut (spre deosebire de data nasterii)
- ora\_inceput, ora\_sfarsit (VARCHAR2(5)): orele proiectiei in format HH:MM. + constrangere list of values in care am adaugat pt ora de inceput de la 8:00 la 21:00 si pentru ora de sfarsit de la 10:00 la 23:00 (default 12:00, respective 14:00)
- tip\_proiectie (VARCHAR2(4)): fiecare film poate avea un tip de proiectie ales din constrangerea de tip list of values. Optiunile sunt cele clasice pentru filme, 2D, 3D, 4DX, si IMAX
- id\_film, id\_sala (NUMBER): referinte catre film si sala. (FK)

## **Tabelul Rezervare**

- id\_rezervare (NUMBER(8)): identificator unic rezervare cu autoincrement. (PK)
- numar\_locuri (NUMBER(3)): numarul de locuri rezervate; + ca val sa fie mai mare ca 0, minim un loc rezervat (default 1)
- locuri\_rezervate (VARCHAR2(7)): lista locurilor rezervate; format R1-C1. Are un check ca locurile sa respecte acest format R1-C1 (Maxim 7 caractere fiindca nu pot avea randul sau coloanal >99, deci maxim pot avea sa zicem R15-C12)
- data\_rez (DATE): data rezervarii. + trigger ca data sa fie in viitor
- status\_rezervare (CHAR(8)): status REZERVAT sau ACHITAT. + list of values doar cu aceste 2 optiuni, pentru a se cunoaste daca biletul este deja achitat cand ajunge clientul (default REZERVAT)
- id\_proiectie, id\_client (NUMBER(8)): referinte. (FK)

## **Tabelul Feedback**

- id\_feedback (NUMBER): identificator unic feedback care are autoincrement. (PK)
- Recenzie (VARCHAR2(250)): (optional) textul recenziei; lungime suficienta pentru comentarii detaliate. + ca lungimea sa fie mai mare ca 0
- data\_recenzie (DATE): data scrierii recenziei. + ca recenzia sa fie in trecut, nu in viitor
- Rating (NUMBER(2)): punctaj acordat experientei + list of ranges de la 1 la 10
- id\_rezervare (NUMBER(8)): referinta la rezervare. (FK)