

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 tablele

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, în care au loc zilnic proiecții de filme, și care permite utilizatorilor să vizualizeze cinematografele din apropiere, recenziile din experiența altor utilizatori la cinema, cele mai noi filme în toate formatele disponibile, orele și zilele de proiecție, să își creeze un profil și să efectueze rezervări fie prin achitarea biletului online, fie la intrarea în sala, înainte de începerea filmului.

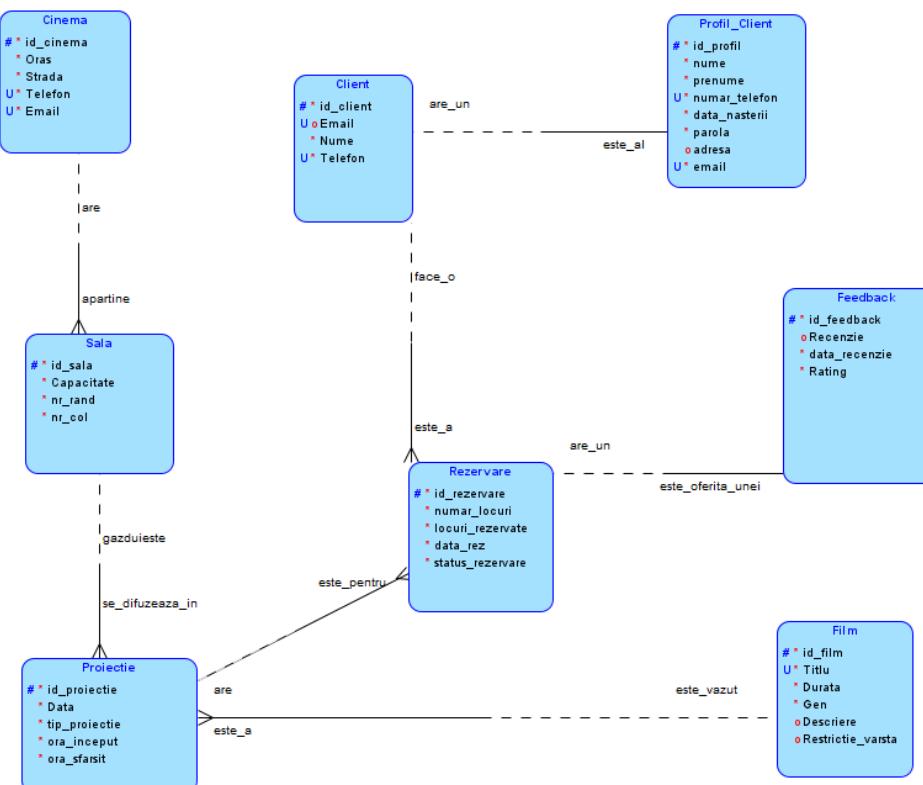
Structura bazei de date este organizată prin tabelele aferente, continând toate datele necesare pentru o funcționare simplă și sigură, care permite să loc de îmbunătățiri și noi atribuții sau tabele (de exemplu: o tabelă pentru barul de snacks-uri și băuturi, în care clientul are la dispozitie un meniu cu suc, popcorn, nachos sau dulciuri).

Ca administrator al bazei de date, pot vizualiza popularitatea fiecarui cinematograf în funcție de cat de frecventat este de către clienți zilnic, pot actualiza lista cu proiecțiile filmelor în momentul în care apare un film nou sau unul rulează de prea mult timp, pot modifica detalii despre cinematografe (cum ar fi numarul salilor atunci când se deschide una nouă sau una este în renovare) și pot consulta rezervările realizate de clienți, inclusiv locurile deja ocupate.

Inspiratia proiectului a venit din franciza Cinema City și modul în care aceasta gestionează experiența utilizatorului pe site. Sistemul oferă informații utile clientilor și facilitează realizarea rezervărilor pe baza datei calendaristice și a unui interval orar, permitând vizualizarea locurilor pe care doresc să le rezerve, folosind o interfață grafică.

O motivare a alegerii acestui proiect este faptul că am realizat un proiect individual, constând într-o aplicație ce poate fi utilizată din perspectiva unui client, dar și din cea a administratorului (server). Din perspectiva clientului, aplicația permite, prin intermediul unei interfețe grafice, alegerea zilei și orei dorite pentru vizionarea unui film, vizualizarea locurilor ocupate și disponibile, iar în urma efectuării unei rezervări, pe baza unui email, stocând detaliile fiecărei rezervări într-o bază de date cu informații utile despre rezervarea clientului, acesta primind pe email o confirmare și având posibilitatea de a anula rezervarea. Ca administrator, pot observa în log-uri actualizările tabelei și pot modifica orice detaliu al informațiilor din interfața grafică.

Proiectul are în vedere corectitudinea datelor inserate în tabele, fiecare tabel având verificări care impiedică introducerea unor valori eronate. Structura bazei de date este construită astfel încât tabelele să fie legate între ele în mod logic, prin chei primare și strângătoare.



Structura si inter-relationarea tabelelor 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) tablei 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)

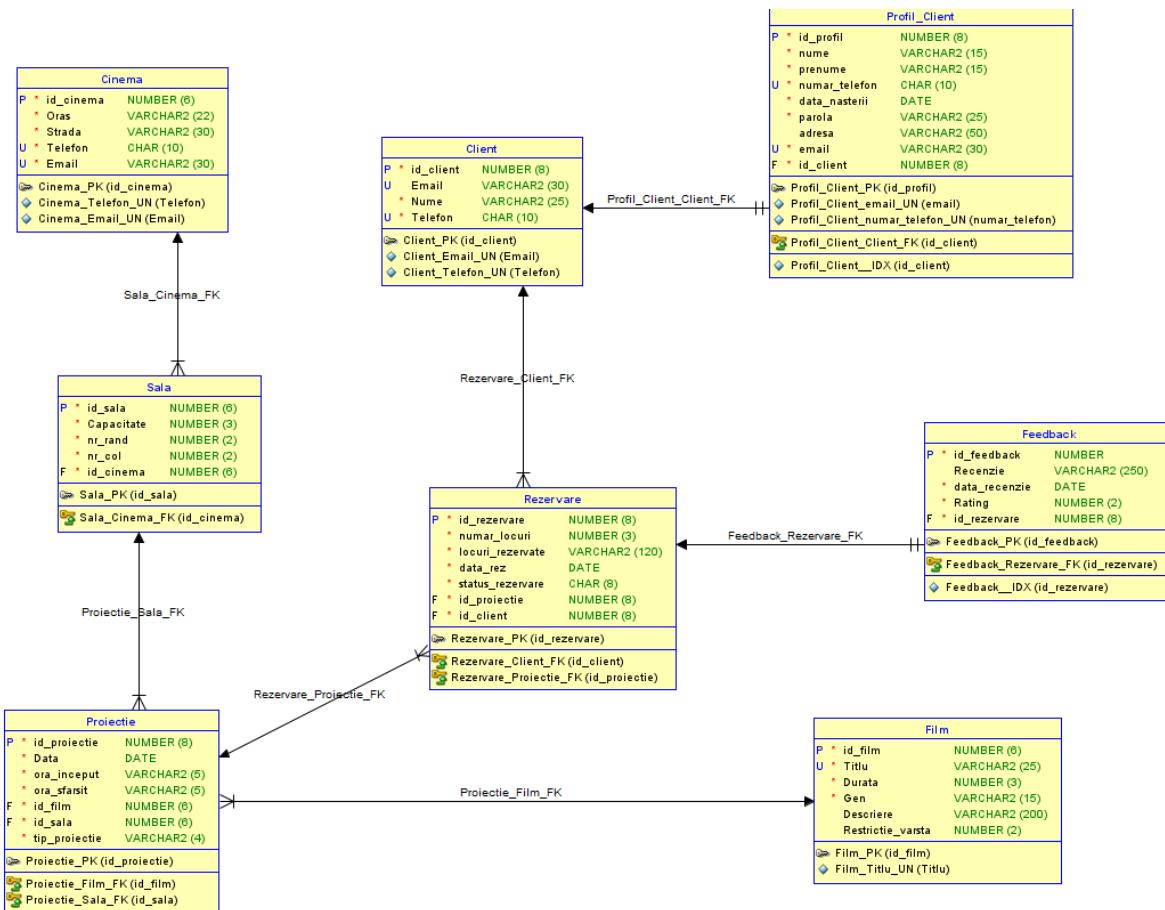
Client->Rezervare (Un client poate face mai multe rezervari)

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, relatiiile sunt logice, dupa cum am discutat anterior.



Prima lege a normalizarii spune ca nu trebuie sa existe coloane repetitive, sau grupuri de atribute 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 depind doar de cheia primara si sa respecta deja prima lege. Daca am fi avut o cheie primara compusa, atunci nu am fi fost in 2NF (decat daca atributele depindeau de intreaga cheie primara compusa), deoarece unele atribute 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 depindă deloc de nicio alta coloana non-cheie. In cazul tabelelor si atributelor din acestea, nu există nicio astfel de situatie. Un contra-exemplu ar fi fost daca in tabela Proiectie aveam si atributele din Film, dar fiindca atributele 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 secenta 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 sevență auto-increment de asemenea. (PK)
- Email (VARCHAR2(30)): optional, dar unic; lungimea de 30 este standard pentru adrese de email uzuale. + constrângere să aibă tipul unui email, format cu '@' și '.'
- Nume (VARCHAR2(25)): nume client; 25 caractere suficient pentru persoane care au chiar și 2 prenume. + constrângere check ca lungimea să fie mai mare ca 1
- Telefon (CHAR(10)): număr de telefon mobil; CHAR pentru lungime fixă. + constrângere să aibă formatul unui număr de telefon din RO, să înceapă cu 07 și poate conține cifre între 0 și 9

Tabelul Profil_Client

- id_profil (NUMBER(8)): identificator unic al profilului cu autoincrement. (PK)
- nume, prenume (VARCHAR2(15)): lungime aleasă pentru nume obisnuite. + constrângere check ca lungimea să fie mai mare ca 2
- numar_telefon (CHAR(10)): telefon profil, fix la 10 caractere. + constrângere să aibă formatul unui număr de telefon din RO, să înceapă cu 07 și poate continua cifre între 0 și 9
- data_nasterii (DATE): data nasterii clientului; tip DATE pentru operații ulterioare de varsta. + trigger ca data nasterii să nu fie în viitor
- parola (VARCHAR2(25)): parola utilizator; lungime suficientă pentru securitate. + constrângere check ca lungimea să fie mai mare ca 8
- adresa (VARCHAR2(50)): adresa completă, optională; lungime mai mare pentru detalii aditionale.
- email (VARCHAR2(30)): adresa email asociată profilului. + constrângere să aibă tipul unui email, format cu '@' și '.'
- id_client (NUMBER(8)): referință 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 filme
- 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 recenzie 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)