Universitatea Tehnica ,,Gheorghe Asachi''din Iasi
Facultatea de Automatica si Calculatoare
Calculatoare si Tehnologia Informatiei (CTI)

Acum îți programezi viitorul!

PROIECT SEMESTRIAL LA DISCIPLINA BAZE DE DATE

Gestiunea activității unui cinematograf

Coordonator:

Cătălin Mironeanu

Student:

Ichim Paula-Mădălina 1308A

Iași ,2020

Titlu proiect:Gestiunea activității unui cinematograf

Analiza, proiectarea și implementarea unei baze de date si a aplicației aferente care să modeleze activitatea unui cinematograf cu privire la gestiunea proiecților de filme și a vânzărilor de bilete.

Descrierea cerințelor si modul de organizare al proiectului

Volumul mare de informaţii existente în cadrul unui cinematograf cu numerosi clienţi dar și numeroase ecranizări determină necesitatea fluidizării fluxurilor de date ce traversează sistemele informaționale, gestiunea acestora fiind o adevărată provocare. Activitatea de gestiune a unui cinematograf implică o muncă intensă în ceea ce priveste organizarea proiecțiilor,a sălilor în programul de lucru al cinematografului,a vinderii de bilete relaționat cu disponibilitatea locurilor în sala în care are loc proiecția filmului.

Informațiile de care avem nevoie sunt cele legate de :

-**cinematograf**: ne interesează ca publicul țintă să cunoască date despre cinematograf în cazul în care aceștia doresc să devină clienți ,să achiziționeze bilete pentru a putea viziona în timpul liber un film cu prietenii sau familia. Din informațiile cu importanță fac parte numele cinematografului, adresa, numărul de telefon și programul aceastuia de lucru.

- **clienți**: datele referitoate la clienti cum ar fi un id specific pentru a putea contabiliza numărul de clienți ,numele acestora și numărul de telefon în cazul în care apar situații neprevăzute ca amânarea unei proiecții sau schimbarea sălii în care are loc proiecția la care aceștia și-au achiziționat bilete.

- **filme**:informațiile despre filme vor fi stocate in două tabele diferite ,va exista o entitate numită **film** ce va conține un id pentru fiecare film ,numele filmului și a regizorului ,acestea vor avea ca și constrângere o cheie unică pentru că nu există un film cu același nume și același regizor,singura excepție sunt filmele care au mai multe părți dar la care este menționat în titlu partea din care face parte,neproducându-se astfel nicio neclaritate.O altă informație utilă stocată in tabela aceasta va fi prețul filmului.A doua entitate care se ocupă de gestiunea filmelor este aceea de **detalii_film**,tabelă aflată în relație de 1:1 cu tabela anterioară,continând informații despre durata filmului,genul filmului și formatul de ecranizare,cum ar fi 2D sau 3D.

-**săli**:cinematograful deține un număr de săli în care au loc ecranizările filmelor,fiecare sală va avea un număr unic reprezentat de un id,pentru o mai ușoară ghidare a clienților cu privire la săli acestea prezintă și un nume care este scris la intrarea în fiecare sală pentru evita confuziile în găsirea locului unde va avea loc proiecția,alte informații utile vor fi numărul de rânduri și scaune,fiecare rând prezintă un număr egal de scaune,sala având o formă dreptunghiulară. O altă entitate ce ne va fi de folos în gestionarea sălilor este cea de "**locuri-săli**" ce conține un id pentru fiecare loc din fiecare sală urmat de rândul și scaunul de pe acel rând ,locuri ce vor fi ocupate de clienți în timpul proiecției iar datele de pe biletul cumpărat vor fi asociate cu un loc din sală.

-**proiecții:**fiecare proiecție va fi identificată de un id_proiecție,deoarece un film poate fi proiectat în mai multe zile și în mai multe săli,așadar proiecțiile vor fi corelate cu sălile și filmele existente în cinematograf.De asemenea se va preciza data și ora la care va avea lor proiecția filmului.

-**vânzări**:pentru fiecare zi se vor înregistra fiecare tranzacție ce constă în achizitionarea de bilete pentru o proiecție,fiecare vanzare va avea un număr unic de id_vânzare.Pe fiecare bilet va fi precizat filmul ,sala și locul corespunzător sălii,ora și data proiectiei astfel tabela de vânzări va avea legături cu majoritatea tabelelor din baza de date.

Descrierea functionala a aplicatiei

Principalele funcţii care se pot întâlni într-un cinematograf sunt:

- Evidenţa operațiunilor referitoare la proiectarea filmelor în săli

- Evidenţa clienţilor

-Evidenţa tranzactiilor efectuate

Descrierea detaliată a entităților și a relațiilor dintre tabele

Tabelele din aceasta aplicație sunt:

-cinematograf;

-sali;

-locuri_sali;

-proiectie;

-film;

-detalii_film;

-vanzari_bilete;

-clienti;

În proiectarea acestei baze de date s-au identificat tipurile de relatii 1:n, n:1 , 1:1 și m:n.

Între tabelele cinematograf si săli se intâlnește o relație de tip one-to-many deoarece un cinematograf poate avea mai multe săli în care să se desfășoare activitatea de proiectare a filmelor. Reciproca însă nu este valabilă, deoarece este imposibil ca o sală să aparțină mai multor cinematografe ,aceasta fiind reprezentată de o incăpere.Legătura dintre cele două tabele este realizată prin câmpul nume_cinematograf.

Între săli si locuri_sali se stabilește o legătura de tip one-to-many. Tabela locuri_sali descrie toate locuri apartinând sălii puse la dispoziția clienților, in timp ce tabela săli prezintă număr de rânduri și scaune ,care înmulțite dau numărul de locuri . Legătura dintre cele două tabele este facută de câmpul id_sala, tabela părinte fiind sala deoarece tabela locuri_sali ajuta la gestiunea ocupării unei săli ,astfel pentru fiecare sală se va dispune de mai multe locuri,însă un loc ,ce este reprezentat de un id unic aparține de o singură sală.

Între săli si proiecție se stabilește o legătura de tip one-to-many. Legătura dintre cele două tabele este facută de câmpul id_sala, tabela părinte fiind proiecție deoarece tabela săli ajută la plasarea spațială a unei proiecții ,astfel pentru fiecare proiecție se va dispune de mai multe săli pentru că aceasta se poate proiecta de mai multe ori în aceeași zi dar și in zile diferite,însă o proiecție,ce este reprezentată de un id unic poate fi ecranizată doar intr-un singur reper spațial, într-o singură sală.

Între film si proiecție se stabilește o legătura de tip one-to-many. Legătura dintre cele două tabele este facută de câmpul id_film, tabela părinte fiind film deoarece tabela proiecție programează rularea unui film la o anumită oră și la o anumită dată,astfel la fiecare proiecție se va ecraniza un singur film,insă filmul se poate proiecta(are mai multe proiecții) de mai multe ori în aceeași zi dar și in zile diferite.

Între film si detalii_film se stabilește o legătura de tip one-to-one. Legătura dintre cele două tabele este facută de câmpul id_film, tabela părinte fiind film deoarece tabela detalii_film enunță detalii despre filmul cu un anumit id și totodată fiecare film in parte conține detalii diferite.

Între vânzări_bilete si locuri_sali se stabilește o legătura de tip one-to-many. Legătura dintre cele două tabele este facută de câmpul id_loc, tabela părinte fiind locuri_sali deoarece tabela vanzari_bilete realizează tranzacția de vindere a biletelor dintr-o sală în care este programată o proiecție,se pot realiza mai multe tranzacții cu privire la un loc dintr-o sală pentru că în săli se produc mai multe proiecții și de asemenea sălile sunt folosite zilnic,însă pe mai multe bilete nu poate apărea același id_loc la aceeași proiecție,pentru că nu se pot așeza mai mulți client concomitent pe un scaun.În schimb fiecare bilet va avea înscripționat pe el ,un singur loc.

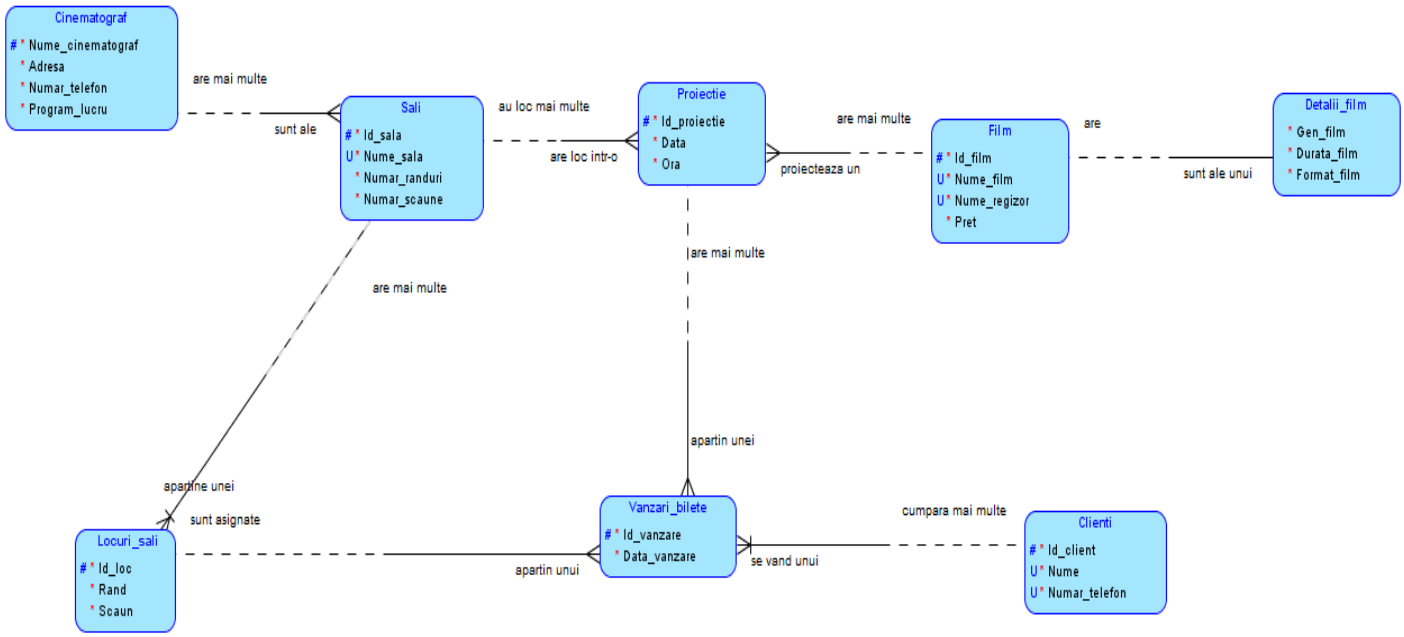
Între vânzări_bilete si proiectie se stabilește o legătura de tip many-to-one . Legătura dintre cele două tabele este facută de câmpul id_proiecție, tabela părinte fiind proiecție deoarece tabela vânzări_bilete realizează vinderea biletelor asignate unei proiecții ce va avea loc,la o proiecție se realizează mai multe tranzacții de vânzare bilete în limita

numărului de locuri din sala unde are loc proiecția,insă un bilet achiziționat este valabil și poate fi folosit doar pentru vizionarea unei proiecții.

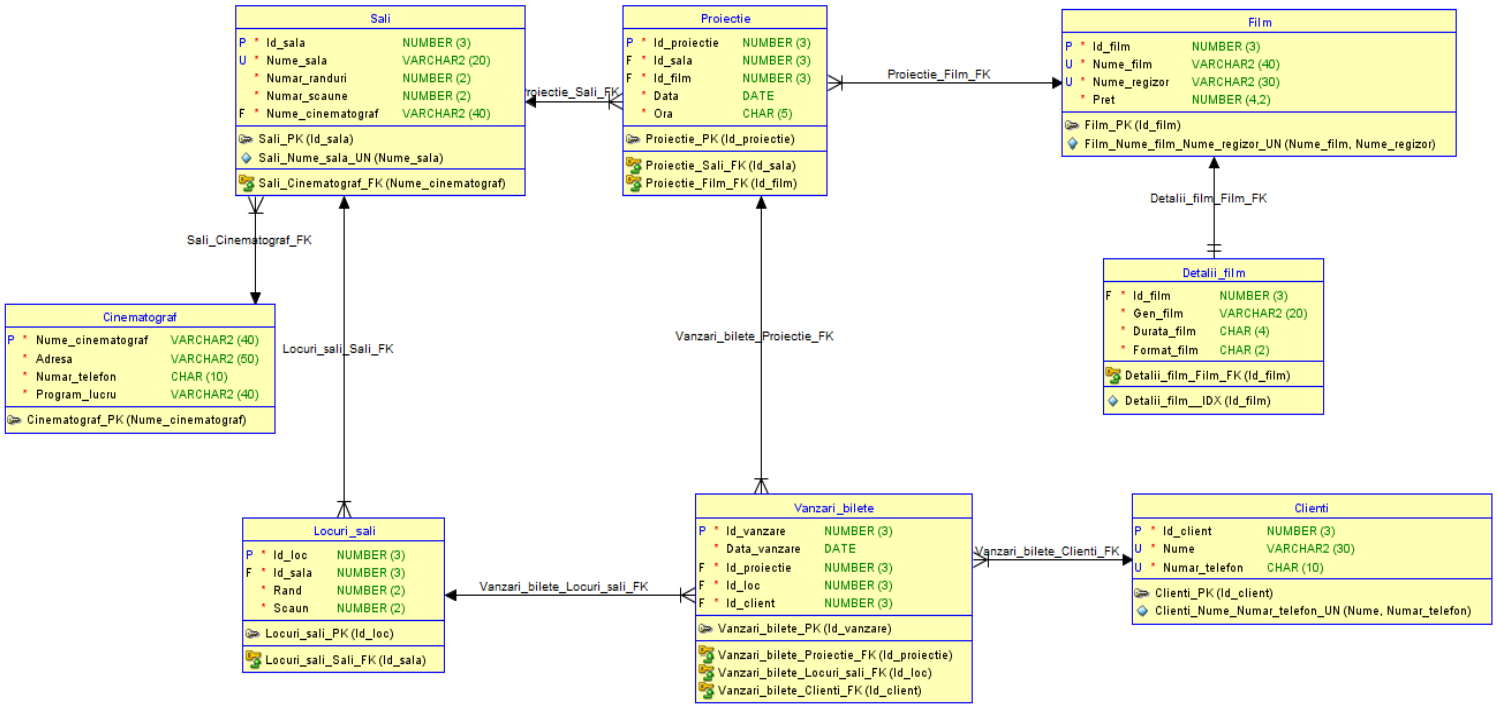
Între clienți si vânzări bilete se stabilește o legătura de tip one-to-many. Legătura dintre cele două tabele este facută de câmpul id_client, tabela părinte fiind client deoarece tabela vânzări are rolul de a înregistra fiecare tranzacție de cumpărare de bilete a unui client,un client poate cumpăra un bilet la un film dar și mai multe în aceeași zi sau poate cumpăra mai multe bilete în decursul de mai multe zile fiind un client fidel al cinematografului, însă o vânzăre de bilete ce are asignată un id_vânzare poate fi doar întocmită doar pentru un client.

De asemenea putem spune că există și relații many-to-many,dar care au fost împărțite în mai multe relații one-to-many/many-to-one ,așa cum este în cazul tabelelor film și săli,unde avem o tabelă de intersecție și anume cea de proiecții.Un film poate avea loc în mai multe săli,acesta fiind proiectat chiar și în aceeași zi de mai multe ori sau în zile diferite.Într-o sală pe parcursul programului de lucru se pot proiecta mai multe filme ,cu o pauză de câteva minute între ele,timp necesar pentru plecarea clienților și pregătirea sălii.

Putem vizualiza mai jos modelul logic la aplicației:



Putem observa de asemenea și modelul relațional .



Constrângeri folosite în cadrul bazei de date

În proiectarea acestei aplicații s-au identificat tipurile de constrângeri de tipul cheie primară(Primary Key),cheie externă(Foreign Key),Unique,Check,Not Null.

Primary key:

Coloana nume_cinematograf din tabela cinematograf reprezentată de un șir de caractere de maxim 40 caractere, prezintă o constrângere de tipul PRIMARY KEY, este o coloană care identifică în mod unic fiecare rând al tabelului.Nici o coloană care face parte din cheia primară nu poate conține valoarea nulă sau să conțină același valori de-a lungul ei.

Coloanele id_sali,id_proiectie,id_film,id_loc,id_vanzare,id_client din tabelele sali, proiectii, film, locuri_sali, vanzari bilete, clienti sunt reprezentate de id-uri numerice de trei cifre create cu ajutorul autoincrementului,astfel evitându-se introducerea de valori egale ,aceste coloane identifică în mod unic fiecare rând al tabelului,nici o coloană care face parte din cheia primară nu poate conține valoarea nulă sau să conțină același valori de-a lungul ei.Fiecare tabel poate conține doar o cheie primară sau să nu aibă nicio cheie primară (cum e in cazul tabelei detalii_film).

Foreign Key:

Constrângerea FOREIGN KEY defineste o coloană sau o combinație de coloane ca foreign key si stabilește o relație între o cheie primară și una unică în același tabel sau în tabele diferite. O valoare care apare intr-un tabel trebuie să se regasească și în cel de-al 2-lea tabel, pe coloana unde formează cheia primară. Constrângerile de tip FOREIGN KEY pot fi definite la nivel de coloană sau tabel.

În cazul nostru toate cheile primare devin chei străine în alte tabela cu excepția cheii primare id_vanzare,acest lucru este datorat relațiilor definite între entități.Cheia primară nume_cinematograf din tabela cinematograf devine cheie străină în tabela săli,datorată relații one-to-many dintre entități.Cheia primara id_sali din tabela sali devine cheie străină în tabelele proiecție și locuri_sali,datorată relațiilor one-to-many. Cheia primara id_proiectie din tabela proiectie devine cheie străină în tabela vanzari_bilete,datorată relației one-to-many. Cheia primara id_client din tabela client devine cheie străină în tabela vanzari_bilete,datorată relației one-to-many.

Not Null:

Constrângerea de tip NOT NULL asigură faptul că o coloană să nu conțină valoarea nulă. Ea poate fi specificată la nivel de coloană și nu la nivel de tabel.

În cadrul aplicației toate coloanele au aplicate această constrângere ,de asemenea este de menționat faptul că prin definiție cheile primare nu pot avea valori nule,așa că atunci când este definită o cheie primară este respectată și condiția de diferit de null,iar cheile străine (chei primare din alte tabele ,in cazul nostru) respectă și ele .

Unique:

O constrângere de integritate de tip cheie unică cere ca fiecare valoare din coloană sau din mulţimea de coloane să fie unice – două înregistrări ale tabelului nu pot avea valori duplicat corespunzătoare cheii unice. Coloana (mulţimea de coloane) inclusă în definiţia cheii unice se numeşte cheie unică. Dacă o cheie unică conţine mai multe coloane se numeşte cheie unică compusă.

Baza de date prezentată are în alcătuirea sa o cheie unică și două chei compuse.Cheia unica este reprezentată de coloana nume_sala din cadrul tabelei sali,in care fiecare sală trebuie să aibă o denumire diferite pentru a putea fi de folos în localizarea unde are loc proiecția de către clienți.Cheile compuse sunt formate din dubletele nume_film și nume_regizor din tabela film,pentru că nu există un film cu același nume și același regizor,singura excepție sunt filmele care au mai multe părți dar la care este menționat în titlu partea din care face parte,neproducându-se astfel nicio neclaritate,al doilea dublet este cel din tabela clienți format din nume_client și număr_telefon neexitând două persoane cu același nume și același număr de telefon.

Check:

Constrângerea de tip CHECK defineşte o condiţie ce trebuie îndeplinită de fiecare linie dintr-un tabel.

Constrângerile utilizate în proiectarea aplicației sunt următoarele:

-prețul unui film să fie mai mare ca 0(pret>0),

-numărul de telefon din cadrul tabelei cinematograf dar și in tabela client să fie de 10 numere și să înceapă cu '07': (substr(numar_telefon, 1, 2) LIKE '07' AND length(numar_telefon) = 10)-numarul de telefon este de tipul char(10)
-durata unui film nu trebuie să depășească 3 ore și 59 minute ,iar formatul în care să fie introdusă durata să fie de tipul "ore:minute": (durata_film este de tip char(4))

```
 ( substr(durata_film, 2, 1) = ':'

   AND to_number(substr(durata_film, 3, 2)) >= 0

   AND to_number(substr(durata_film, 3, 2)) <= 59

   AND to_number(substr(durata_film, 1, 1)) >= 0

   AND to_number(substr(durata_film, 1, 1)) <= 3 );
```

-ora la care să fie proiectat un film să aibă formatul "ora:minute": (să se incadreze între 00:00 și 23:59)

```
    ( substr(ora, 3, 1) = ':'

     AND to_number(substr(ora, 4, 2)) >= 0

     AND to_number(substr(ora, 4, 2)) <= 59

     AND to_number(substr(ora, 1, 2)) >= 0

     AND to_number(substr(ora, 1, 2)) <= 23 );
```

-ultima constrângere folosită este asupra unei date calendaristice ,mai exact cea a proiecțiilor deoarece cinematograful are în vedere planificarea proiecțiilor viitoare iar cele petrecute în trecut nu mai au rostul de a încurca fluxul de informații,iar de asemenea nu se mai pot vinde bilete la o proiecție din trecut.

```
IF( :new.Data <= SYSDATE )

THEN

RAISE_APPLICATION_ERROR( -20001,

'Data invalida: ' || TO_CHAR( :new.Data, 'DD.MM.YYYY HH24:MI:SS' ) || ' trebuie sa fie mai mare decat data curenta.' );
```