Fișa Cerințelor

# Titlul Proiectului : “Bază de date pentru o biblioteca”

Realizatori :

Petricioiu Robert - Emanuel, anul II, grupa A5

Necula Ervin, anul II, grupa A5

**Cuprins:**

1. Descrierea proiectului
2. Utilizatori
3. Entități si tabele
4. Descrierea proiectului

Proiectul constă într-o bază de date pentru o biblioteca dintr-o anumită locație. Baza de date va realiza gestiunea bibliotecii, salvarea datelor într-o formă organizată, accesibila și ușor de folosit.

Baza de date va fi modulară pentru a ușura întelegerea si ușurința în utilizare.

Momentan, baza de date a fost concepută ca fiint alcătuită din șase tabele, și anume : o tabelă conținând autorii cărților respective, o tabelă conținând persoanele care sunt au un carnet de bibliotecă, o tabelă conținând toate cărțile deținute de bibliotecă și o altă tabelă în care vom reține împrumurile de cărți .

Adițional, am incorporat și împrumutul de filme, care de asemenea se traduce în două tabele, una pentru filmele deținute de bibliotecă și o altă tabelă care reține imprumuturile.

1. Utilizatori

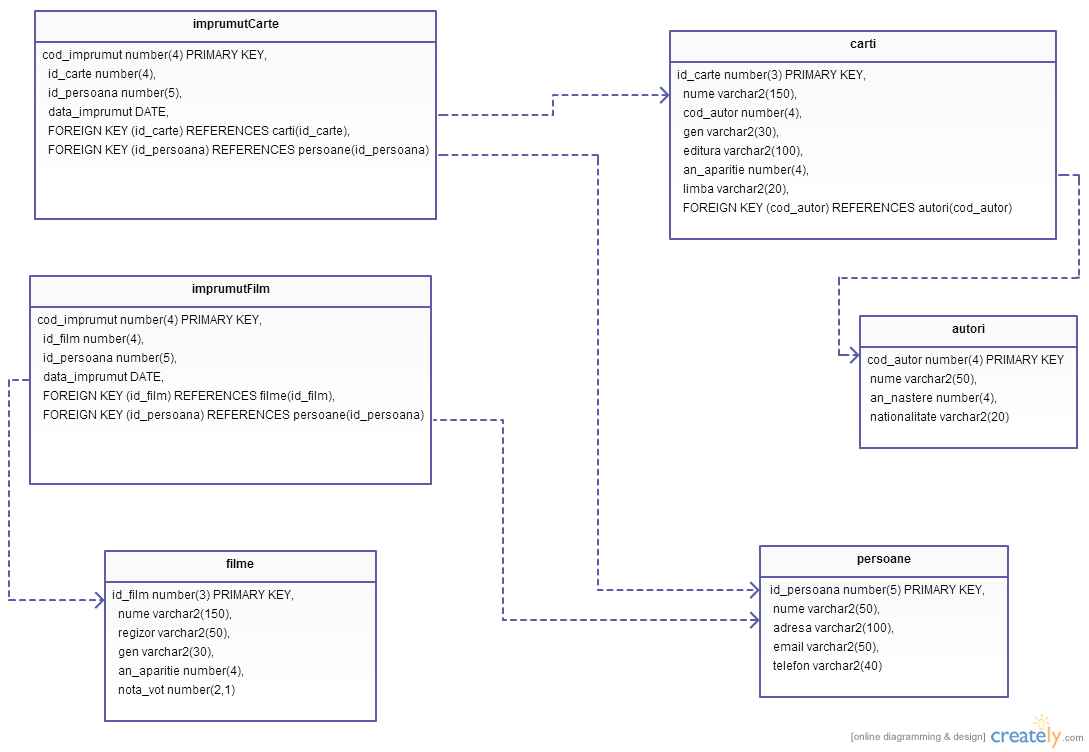
Utilizatorii acestei aplicații ar putea fi administratorii bibliotecii, dar dacă se concepe și o interfață pentru utilizator, ea poate fi folosită și de către client (odată stabilite nivelurile de acces).

1. Entități și tabele

Tabelele folosite pentru realizarea bazei de date (momentan):

1. Autori
   1. cod\_autor (numeric) \*
   2. nume (caractere)
   3. an\_nastere (numeric)
   4. nationalitate (caractere)
2. Persoane
   1. id\_persoana (numeric) \*
   2. nume (caractere)
   3. adresa (caractere)
   4. email (caractere)
   5. telefon (caractere)
3. Carti
   1. id\_carte (numeric) \*
   2. nume (caractere)
   3. cod\_autor numar (\_) referă către codul autorului din tabela de autori
   4. gen (caractere)
   5. editura (caractere)
   6. an\_aparitie (numeric)
   7. limba (caractere)
4. Filme
   1. id\_film (numeric) \*
   2. nume (caractere)
   3. regizor (caractere)
   4. gen (caractere)
   5. an\_aparitie (numeric)
   6. nota\_vot (numeric)
5. ImprumutFilm
   1. cod\_imprumut (numeric) \*
   2. id\_film (numeric) referă către identificatorul din Filme
   3. id\_persoana (numeric) referă către identificatorul din Persoane
   4. data\_imprumut (dată)
6. ImprumutCarte
   1. cod\_imprumut (numeric) \*
   2. id\_carte (numeric) referă către identificatorul din Carți
   3. id\_persoana (numeric) referă către identificatorul din Persoane
   4. data\_imprumut (dată)

(câmpurile marcate cu \* replezinta câmpuri care au valori unice pentru tabela respectivă)



# Explicarea necesitatii elementelor folosite

Am creat proceduri pentru :

* Realizarea unui imprumut de carte
* Realizarea unui imprumut de film
* Crearea unor view-uri cu sql dinamic, contin id-ul persoanei si cartile imprumutate de acesta
* Returnarea unei carti
* Returnarea unui film

Am creat functii pentru:

* Adaugarea unui autor
* Adaugarea unei noi carti
* Adaugarea unui film
* Adaugarea unei persoane
* Eliminarea unei persoane

Am creat un trigger pentru:

* Aruncarea unei exceptii atunci cand utilizatorul imprumuta o carte de mai mult de 3 ori, fiind obligat sa returneze cel putin o carte. Se arunca o exceptie prinsa de interfata

Am creat un materialized view:

* Este folosit de trigger pentru numararea cartilor pentru fiecare user
* Isi da refresh dupa fiecare commit(in general commit-uri pt inserturi)

Am creat exceptii particulare pe care le prind in interfata:

* Utilizatorul imprumuta de doua ori aceeasi carte/acelasi film
* Utilizatorul imprumuta mai mult de 3 carti
* Incercam sa stergem un autor fara a sterge toate cartile autorului
* Incercam sa sterge o carte fara a sterge toate imprumuturile acestei carti
* Incercam sa stergem o persoana care inca are carti imprumutate

Am folosit indecsi pentru cautarile pe care le fac pe tabela, si anume dupa id-urile cartilor/filmelor si pentru cele din view-ul de admin

Implementare diferentiata pentru user/admin prin login diferentiat pentru fiecare.

# Limbajul folosit

Am folosit limbajul java pentru interfata, si un design usor de folosit si de inteles atat pentru user cat si pentru admin.

Permit search pentru carti, nu am implementat pentru filme, implementarea fiind similara.

Am butoane care permit navigarea usoara prin proiect.

# Pe partea de admin

Permite vizualizarea cartilor pe care le detin toti userii si bineinteles, pentru un user in parte.

Permite vizualizarea tuturor cartilor din biblioteca.

Permite adaugarea unei carti la biblioteca, facand din interfata un select max(id\_carte)+1, astfel obtin un id unic pe care il setez in casuta interfetei de adaugare, fiind mai comod.

Permite stergerea unei carti (cu exceptiile mentionate mai sus)

Permite adaugarea de autori, similar ca la carti, precum si stergerea unui autor (cu exceptiile propriile) precum si o vizionare a listei tuturor autorilor.

Permite vizualizarea tuturor filmelor imprumutate a tuturor userilor.

Permite vizualizarea tuturor filmelor din biblioteca precum si adaugarea unui film la Biblioteca, similar ca la carti(cu interfata specializata pentru filme)

Permite operatiuni pentru clientii bibliotecii prin sectiunea (Persoane). Vom permite adaugarea de clienti/eliminarea precum si o vizualizare a tuturor clientilor.

# Motivarea folosirii exceptiilor si trigger-ului.

Folosim exceptii pentru user, astfel nu permitem ca un user sa imprumute mai mult de 3 carti, precum si imprumutarea aceleiasi carti de doua ori. Se va deschide o fereastra modala care indica actiunea pe care trebuie s-o realizeze utilizatorul/adminul.

Avem exceptii si pe parta de user, nu permitem stergerea unui autor, atat timp cat acesta inca detine carti in biblioteca. De asemenea, nu permitem stergerea unei carti daca inca exista un imprumut de carte. Similar este realizat si pentru filme respectiv persoane .

Nu putem sterge un film fara a sterge toate imprumuturile acestui film. Nu putem, de asemenea sa stergem persoane daca acestia inca au imprumuturi si carti nereturnate/filme returnate.

La partea de inserare, nu este permisa inserarea unor elemente. De exemplu avem un RegEx care verifica daca telefonul are un format corect sau un RegEx pentru e-mail sa vada daca acesta are formatul corect