

**PROGRAMAREA INTERFETELOR PENTRU BAZE DE DATE**

**Profesor coordonator**: **Ș. I. Dr. Ing. Pupezescu Valentin**

**STUDENT**: **Angan Lavinia Marilena**

# **GRUPA**: **433A**

**CUPRINS**

## Tema proiectului ………………………………………………………………….3

## Descrierea sistemului de gestiune a bazelor de date MySQL …………………….3

## 3) Tehnologia JSP utilizata in dezvoltarea aplicatiei …………………………………4

## 4) Limbajul HTML si utilitatea sa in aplicatii ………………………………………..5

## 5) Limbajul CSS……………………….…..…………………………………………5

## 6) Descrierea aplicatiei ………………………………………………………………..5

## 7) Concluzii ………………………………………………………………………….18

## 8) Bibliografie ……………………………………………………………………….19

## 1) Tema proiectului

Tema proiectului se bazeaza pe dezvoltarea unei aplicatii ce contine o baza de date, creata in sistemul de gestionare a bazelor de date MySQL. Se pot utiliza diferite tehnologii: JSP, Hibernate, JPA, .NET, Python+Django, Python+Flask, etc.

Interfetele vor trebui sa permita utilizatorului sa execute urmatoarele operatii pe toate tabele: vizualizare, adaugare, modificare si stergere de date. Vizualizarea tabelelor de legatura va presupune vizualizarea datelor referite din celelalte tabele.

Pentru tema individuala primita, am ales 2 tehnologii diferite: JSP si Hibernate. In aceasta prezentare ne vom orienta atentia catre tehnologia JSP. Asocierea pentru tabelele din baza de date este de M:N

## 2) Descrierea sistemului de gestiune a bazelor de date MySQL

**2.1. Ce este MySQL?**

MySQL este un sistem de gestionare a bazelor de date relationale open source care este utilizat in principal pentru aplicatiile online. MySQL poate crea și gestiona baze de date foarte utile (cum ar fi informatii despre angajati, inventar și multe altele), la fel ca alte sisteme, cum ar fi popularul Microsoft Access. In timp ce Microsoft Access, MySQL și alte sisteme de gestionare a bazelor de date servesc scopuri similare (de a gazdui datele), utilizarea difera foarte mult.

MySQL este componenta integrata a platformelor LAMPsau WAMP (Linux/WindowsApache-MySQL-PHP/Perl/Python). Popularitatea sa ca aplicatie web este strans legata de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. In multe carți de specialitate este precizat faptul ca MySQL este mult mai usor de invatat și folosit decat multe din aplicatiile de gestiune a bazelor de date, ca exemplu comanda de iesire fiind una simpla și evidenta: „exit” sau „quit”.

**2.2. Cum administreaza MySQL bazele de date?**

Pentru a administra bazele de date MySQL se poate folosi modul linie de comanda sau, prin descarcare de pe internet, o interfata grafica:MySQL AdministratorșiMySQL Query Browser. Un alt instrument de management al acestor baze de date este aplicatia gratuita, scrisa în PHP,phpMyAdmin.

**2.3. De ce este atat de util MySQL?**

Baza de date MySQL este folosita in principal ca mijloc de a stoca date pentru aplicatii mari, bazate pe web. Site-uri precum WordPress, iStock, GitHub, Facebook, NASA, Marina SUA, Tesla, Scholastic, Spotify, YouTube, Netflix, Glasses Direct, Symantec (si multe altele) folosesc baza de date MySQL ca mijloc de stocare a datelor pe din interiorul sau exteriorul site-urilor web și serviciilor interne.

## 3) Tehnologia JSP utilizata in dezvoltarea aplicatiei

**3.1. Ce este JSP?**

Java Server Pages este o simpla, dar puternica tehnologie folosita pe partea de server pentru a genera continut HTML dinamic. JSP este o extensie directa a Java Servlets si furnizeaza o modalitate de a separa partea de procesare de cea de prezentare. Motorul JSP este doar un alt Servlet, mapat la extensia \*.jsp.

Paginile JSP sunt create sa suporte mai multe tipuri de documente structurate, indeosebi HTML si XML. În general, JSP-urile folosesc anumite informatii pe care le trimit la server intr-o cerere HTTP care interactioneaza cu datele existente pe acesta si creaza dinamic un raspuns organizat intr-un format standard (HTML, DHTML, XML, etc.) sau intr-un format text sau neorganizat ce va fi trimis înapoi clientului.

**3.2. Ce este SERVLET?**

Un servlet este un fișier java care poate solicita clientul, proceseaza și furnizează un fisier HTML ca raspuns. Exista mai multe servlete in interiorul containerului web. De asemenea, este posibila maparea mai multor cereri la un servlet. Prin urmare, toate aceste configuratii sunt incluse in acest fisier special, care este fisierul web.xml.

**3.3. Care este diferenta majora intre JSP si SERVLET?**

Principala diferentaintre JSP și Servlet este caJSP este un limbaj de scripting al paginii web care poate genera continut web dinamic în timp ce servletul este un program Java care este deja compilat și folosit pentru a crea conținut dinamic de web.

Intr-o aplicatie web obisnuita, clientul solicita o pagina Web de la server și serverul raspunde inapoi cu pagina necesara. Aceste pagini pot fi fie statice, fie dinamice. Continutul unei pagini statice este deja creat. Continutul unei pagini dinamice este creat in timpul rularii. JSP și Servlet sunt două metode în Java pentru a crea pagini web dinamice. JSP este tradus si compilat intr-un servlet de catre containerul web. Pe de alta parte, un servlet este un program Java care este gestionat de containerul web.

## 4) Limbajul HTML si utilitatea sa in aplicatii

**4.1. Ce este HTML?**

Unul din primele elemente fundamentale ale WWW ( World Wide Web ) este HTML ( Hypertext Markup Language ), care descrie formatul primar în care documentele sunt distribuite și văzute pe Web. Multe din trasaturile lui, cum ar fi independenta fata de platforma, structurarea formatării și legaturile hypertext, fac din el un foarte bun format pentru documentele Internet și Web.

**4.2. In ce scop utilizam limbajul HTML?**

Scopul HTML este mai degraba prezentarea informatiilor – paragrafe, fonturi, tabele ș.a.m.d. – decat descrierea semanticii documentului. In cadrul dezvoltarii web de tipfront-end, HTML este utilizat impreuna cu CSS și JavaScript.

## 5) Limbajul CSS

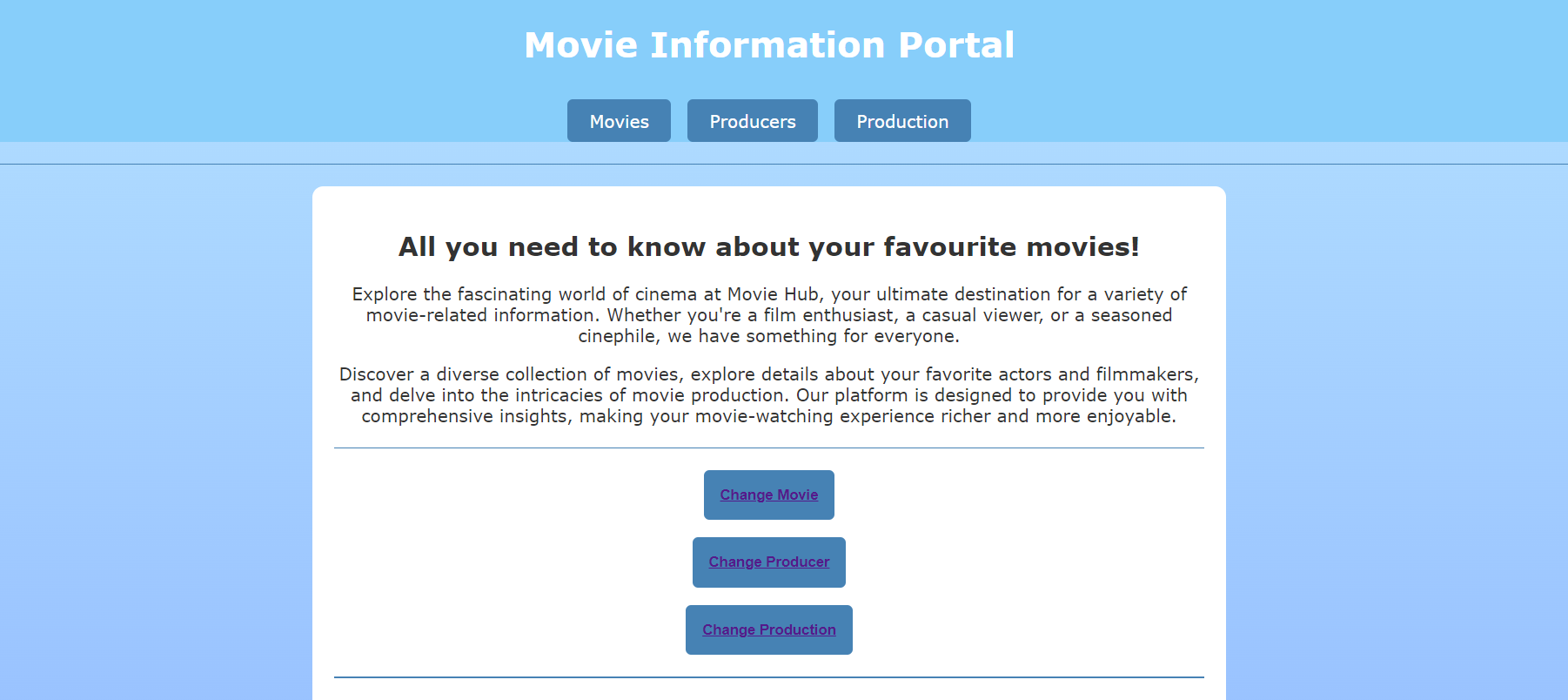
## 

**5.1. Ce este CSS?**

CSS (Cascading Style Sheets) este un limbaj de stil care se utilizează pentru a defini aspectul unei pagini web. Cu ajutorul CSS, se pot specifica elemente precum font, culoare, margini, spațiere, poziționare și multe altele pentru fiecare element din pagina web. CSS este unul dintre cele trei elemente principale ale HTML, HTML fiind limbajul de marcare pentru conținut, CSS fiind limbajul de stil, și JavaScript fiind limbajul de programare pentru interacțiune.

## 

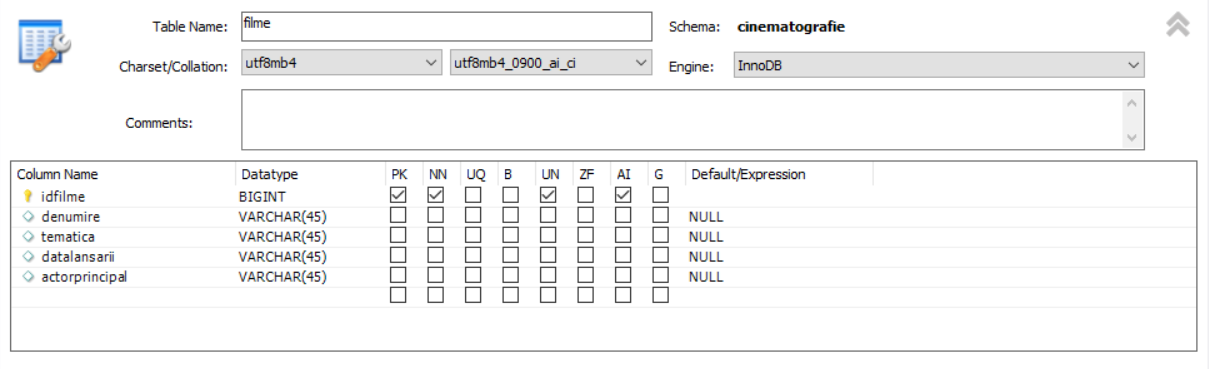
## 6) Descrierea aplicatiei



**6.1. Baza de date**

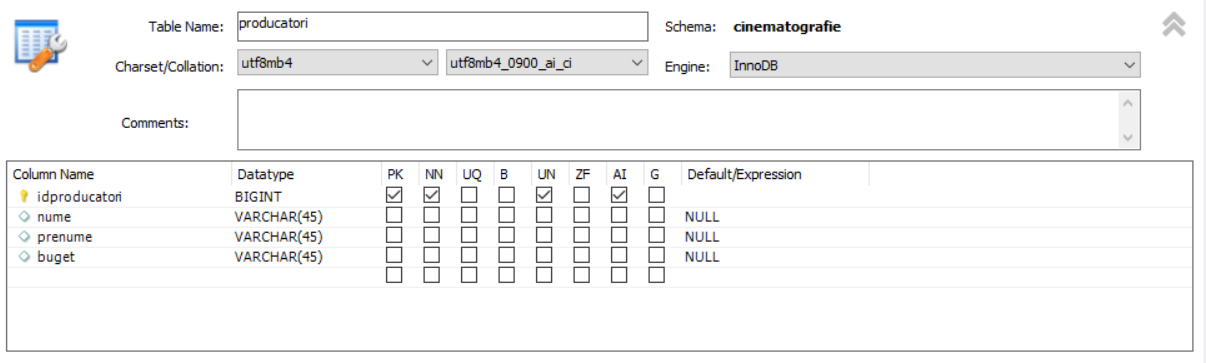
Tema individuala se bazeaza pe crearea unei baze de date ce are 2 tabele in asociere M:N. Tabelele sunt: **filme** si **producatori**. Pentru cele 2 tabele am ales cateva atribute caracteristice:

Pentru tabela **filme**: am ales ca si cheie primara idfilme. Celelalte atribute sunt: denumire, tematica, datalansarii, actorprincipal.



***Figura 1:*** *tabela filme*

Pentru tabela **producatori** am ales ca si cheie primara idproducatori. Celelalte atribute sunt: nume, prenume, buget.



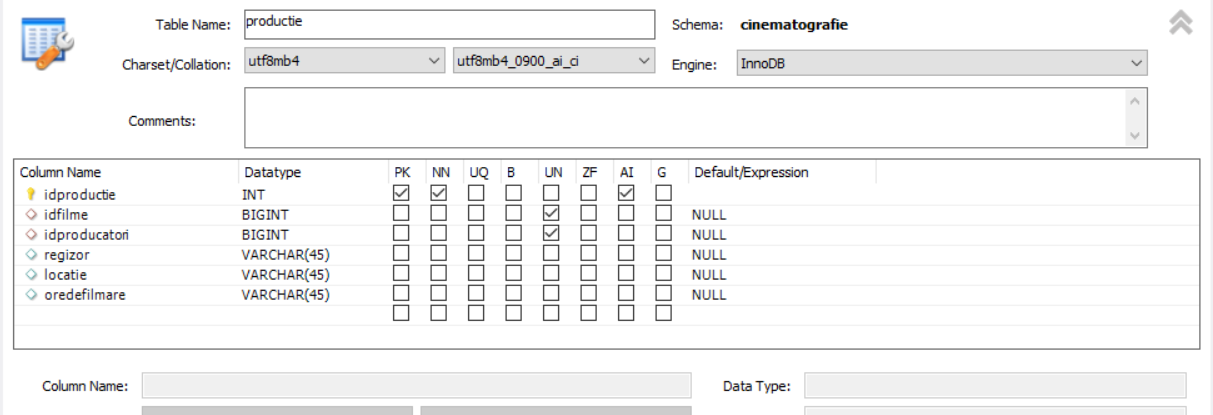
***Figura 2:*** *tabela locatii*

**6.2. Ce este asocierea M:N?**

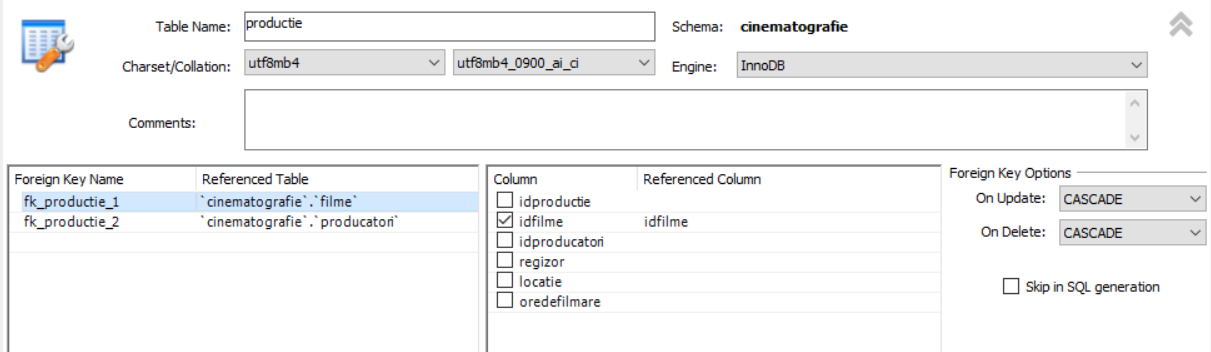
Asocierea M:N este o relație între două tabele într-un bază de date care permite unui element din prima tabelă să aibă mai multe legături cu mai multe elemente din a doua tabelă și invers. Acest lucru se realizează prin intermediul unei tabele intermediere care conține informații despre legăturile dintre elementele din cele două tabele.

**6.3.** **Ce reprezinta tabela de legatura?**

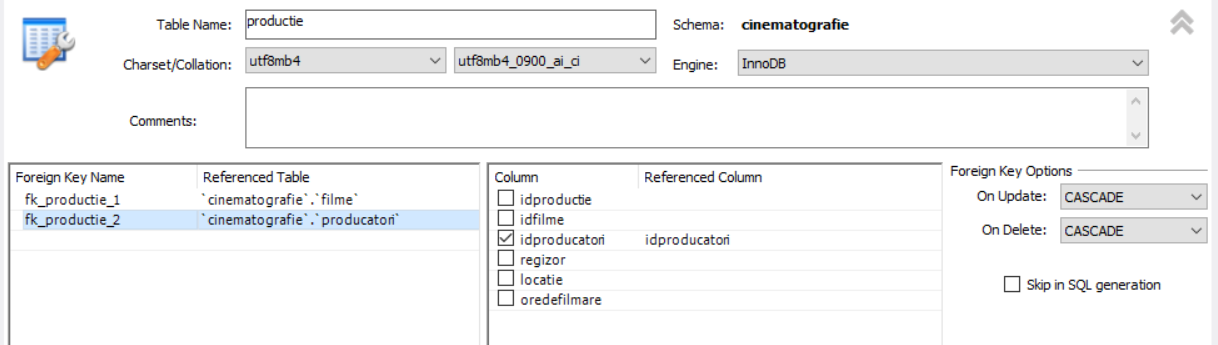
Pentru a crea o relație mai-multi-la-mai-multi, trebuie sa se creeze o a treia tabela denumita deseori tabela de jonctiune, care imparte relatia mai-multi-la-mai-multi în doua relatii unu-la-mai-multi. In cazul nostru, am ales ca si tabela de legatura tabela **productie**. In aceasta noua tabela, atributele ce au fost selectate ca si chei primare pentru tabelele anterioare vor deveni chei straine (FK) pentru tabela de legatura **productie**



***Figura 3:*** *tabela productie*

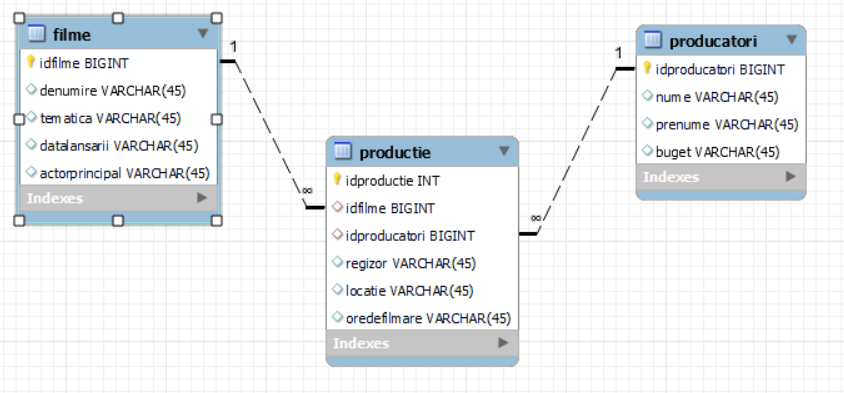


***Figura 4.a:*** *tabela productie foreign keys*



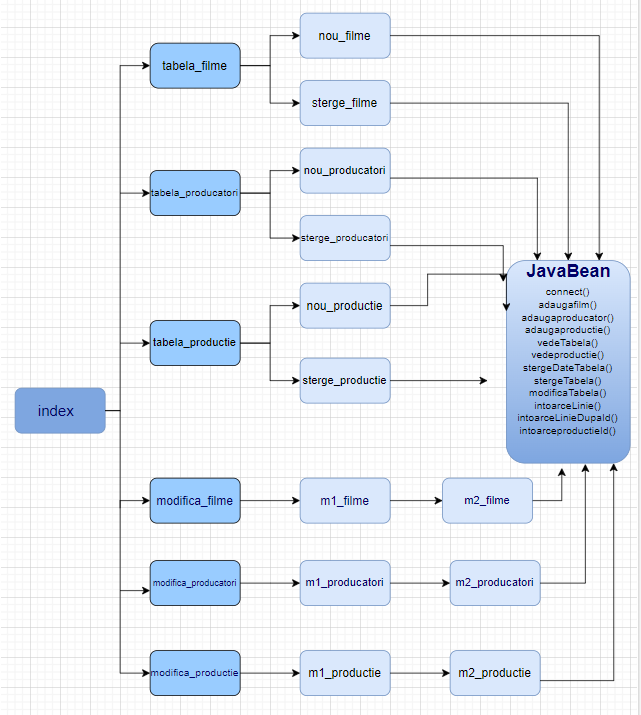
***Figura 4.b:*** *tabela productie foreign keys*

**6.4.** **Diagrama logica a bazei de date (Diagrama ERD)**



***Figura 5:*** *Diagrama ERD*

**6.5.** **Diagrama implementarii fisierelor ce alcatuiesc pagina web**



***Figura 6:*** *Diagrama UML*

**6.6.** **Functionalitatea aplicatiei**

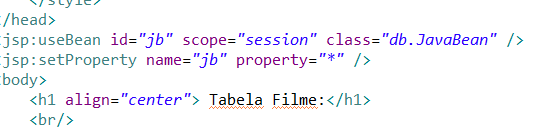
**6.6.1 Arhitectura proiectului**

Proiectul realizat in tehnologia JSP are urmatoarea structura:

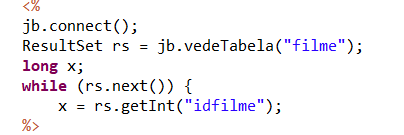
* Un pachet DB ce continue clasa JavaBean, ce are rolul de a oferi toate functionalitatile principale ale interfetei bazei de date si anume: conectica, operatiile de afisare, adaugare, modificare, stergere.
* Folder-ul webapp ce continue toate paginile JSP, ce au rolul de a importa functionalitatile din clasa JavaBean, fiind conectate intre ele si implicit conectate la pachetul DB, ce realizeaza partea dinamica a proiectului.

**6.6.2 Implementarea functiilor**

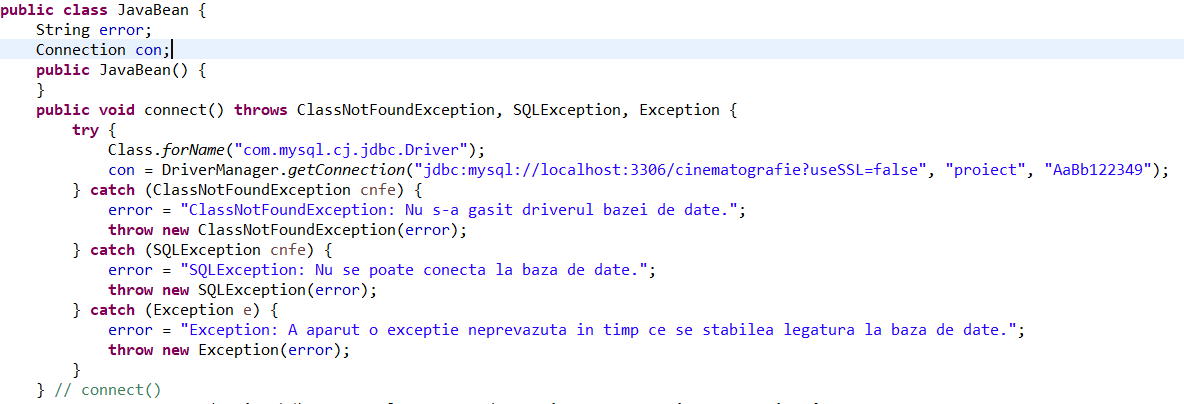
Toate paginile JSP contin structuri de cod java, ce ofera aplicatiei comportamentul dinamic. Ca sa utilizam functiile din clasa java, trebuie sa utilizam tag-urile JSP.



Jb este un obiect instanta la clasa JavaBean, astfel facandu-se legatura directa intre paginile JSP si functiile pe care dorim sa le implementam.



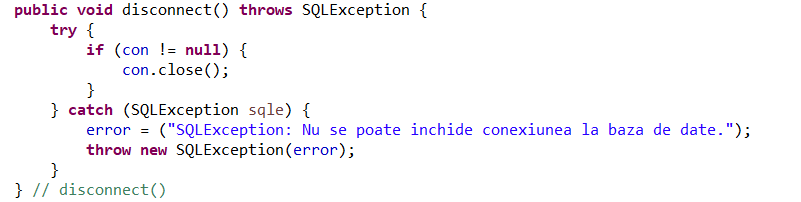
In tabela pe care am dat-o exemplu, am create un formular, in cadrul caruia prin instructiunea jb.connect() ne legam la functia connect() din clasa JavaBean:



Obiectul “con” este cel prin intermediul caruia se realizeaza toate operatiile pe baza de date. In functia connect() : se incarca driver-ul de MySQL. DriverManager contine mai multe drivere de conectica, mai multe SGBD-uri.

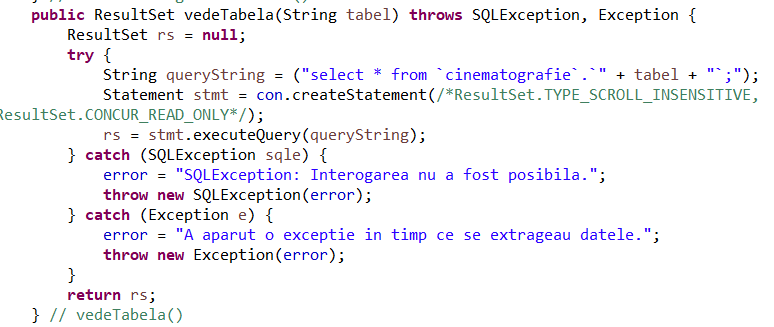
Prin functia getConnection(…) apelam functia de conectare si are ca atribute baza de date realizata in MySQL, user-ul si parola utilizatorului.

Totodata, exista si o functie disconnect() ce realizeaza close() pe obiectul de conexiune con:



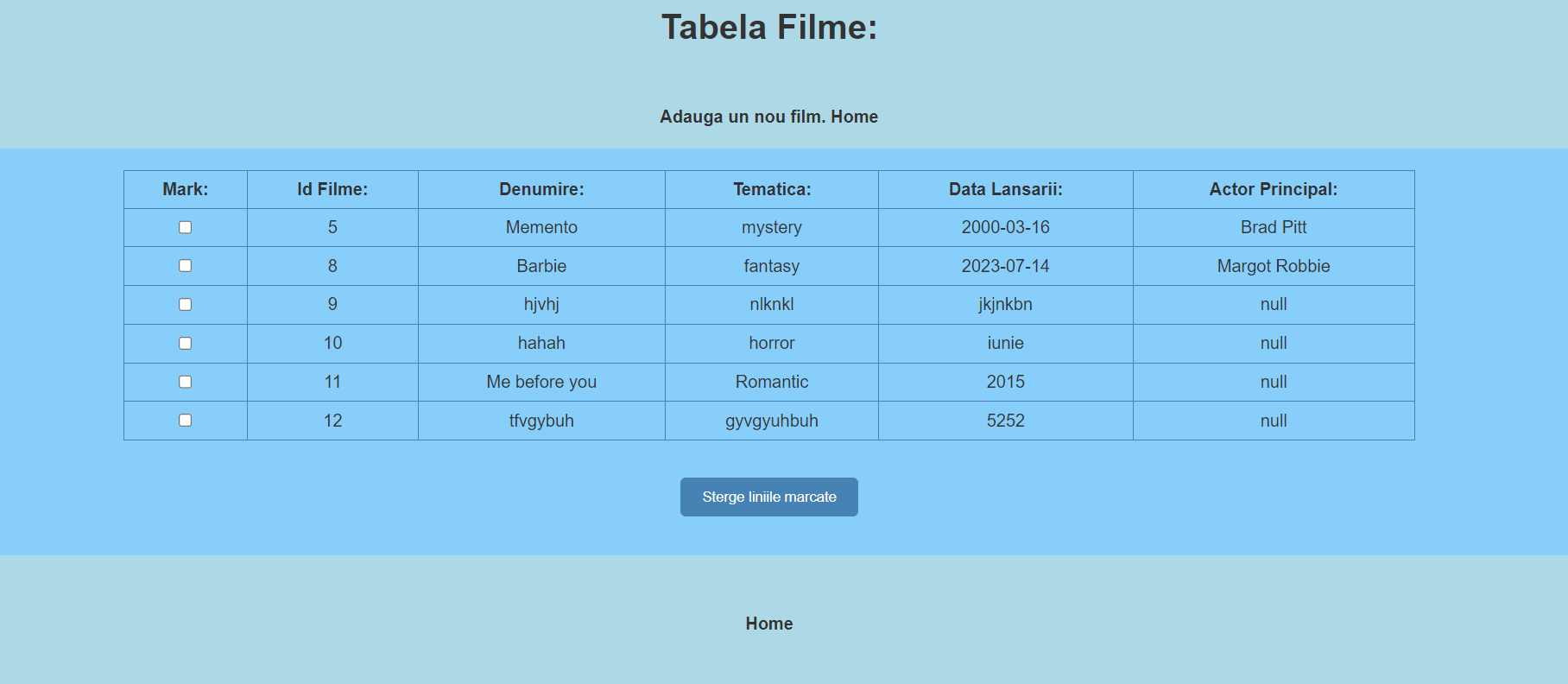
**Cum se afiseaza datele?**

In primul rand alegem ca si exemplu tabela filme. Functia din clasa JavaBean este vedeTabela() ce primeste ca si atribut numele tabelei pe care dorim sa facem interogarea si returneaza un set de rezultate: linii de tip “filme”, ce le implementeaza intr-un obiect de tip results. Acesta este declarat initial null.



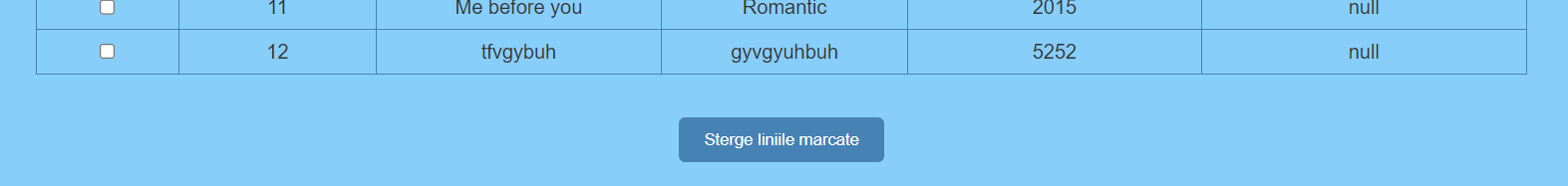


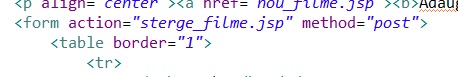
Incepand cu linia 95 cu structura repetitiva while, se plaseaza ‘’cursorul’’ pe prima linie, si se preia din “idfilme” valoarea id-ului ce se leaga de valorile pe care dorim sa le afisam.

Vom obtine prin apasarea butonului **Filme** din cadrul index-ului urmatoarea pagina: 

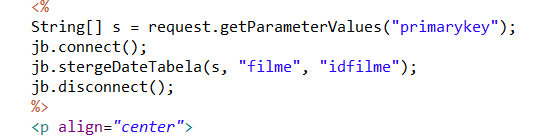
**Cum se realizeaza stergerea datelor?**

Luam drept exemplu stergerea datelor unui gen muzical. Asadar ne indreptam atentia catre fisierul sterge\_filme.jsp.

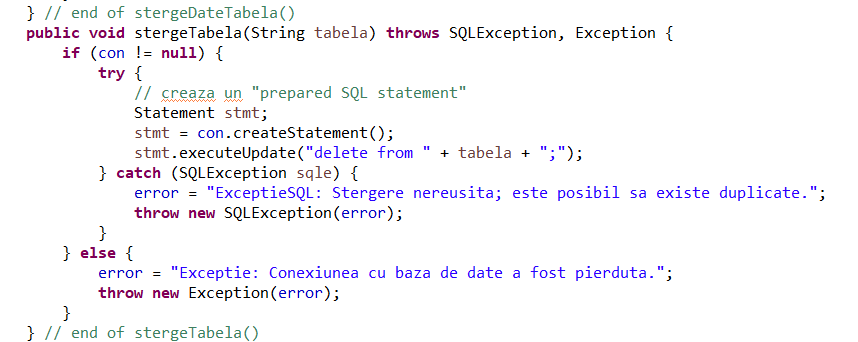




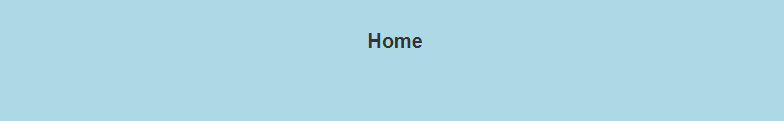
Prin metoda POST, valorile ascunse in spatele checkbox-urilor sunt transmise prin functia getParameter().



Se memoreaza toate datele pe care le-am bifat si se plaseaza in vectorul s, si prin obiectul jb se apeleaza la functia stergeDateTabela(…) din clasa JavaBean, functie ce are ca atribute tabela si totodata campul dupa care se doreste sa se faca stergerea.

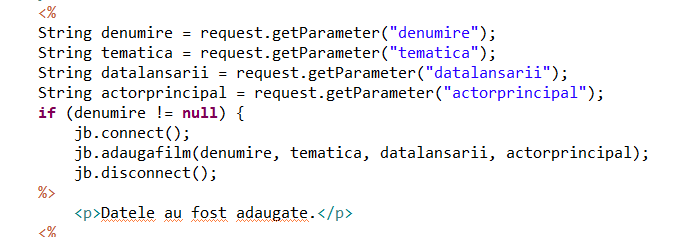


Prin instructiunea if (con != null) se verifica daca avem conexiune cu baza de date. Apoi se declara un obiect “delete” de tip PreparedStatement, declaratie ce se realizeaza pe conexiunea con. Se insereaza instructiunea specifica MySQL urmata de "=?;" ce pune in evidenta faptul ca noi nu cunoastem cate id-uri au fost selectate. Prima valoare ce se ascunde sub “?” se inlocuieste cu valoarea aux, dupa care se executa si apoi se sterge din baza de date intreaga linie. Acest ciclu se continua pana cand se sterg toate datele ce au fost selectate. La finalul operatiei, se va afisa urmatoarea pagina:



**Cum se realizeaza adaugarea datelor?**

Pentru aceasta operatie vom lua ca exemplu tabela de legatura productie. Deci ne concentram atentia asupra fisierului nou\_filme.jsp.

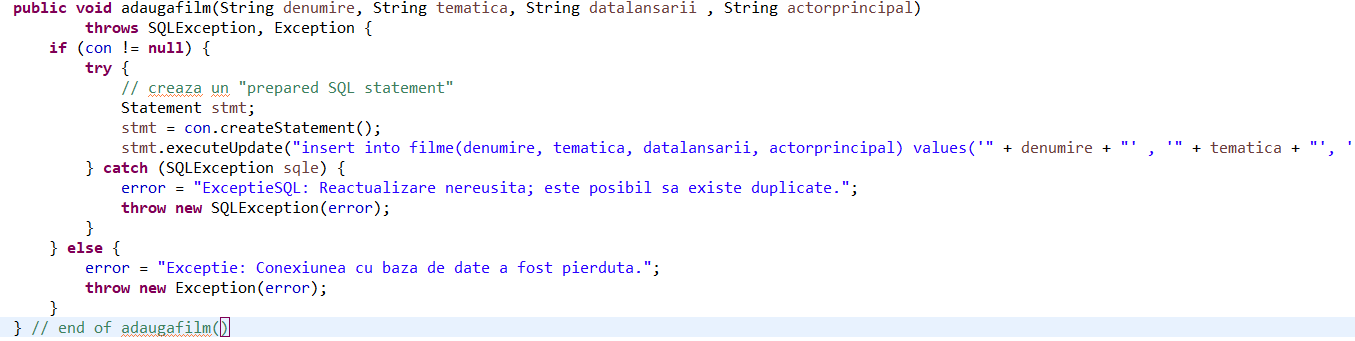




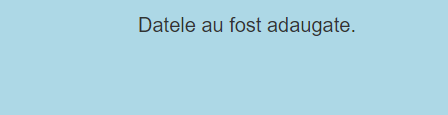
Pagina in cadrul careia se adauga datele este:



O data apasat butonul “Adauga filmul”, se va face legatura cu functia adaugafilm(…) din clasa JavaBean:

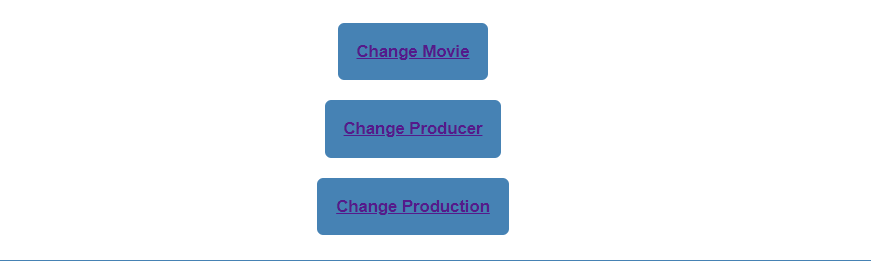


Iar pagina corespunzatoare va fi:



**Cum se realizeaza modificarea datelor?**

Am ales ca pe interfata site-ului , operatia de modificare sa fie separata fata de celelalte. Asadar vom lua exemplu tabela de filme, deci ne vom indrepta atentia catre index, unde am introdus butonul pentru operatia de modificare:



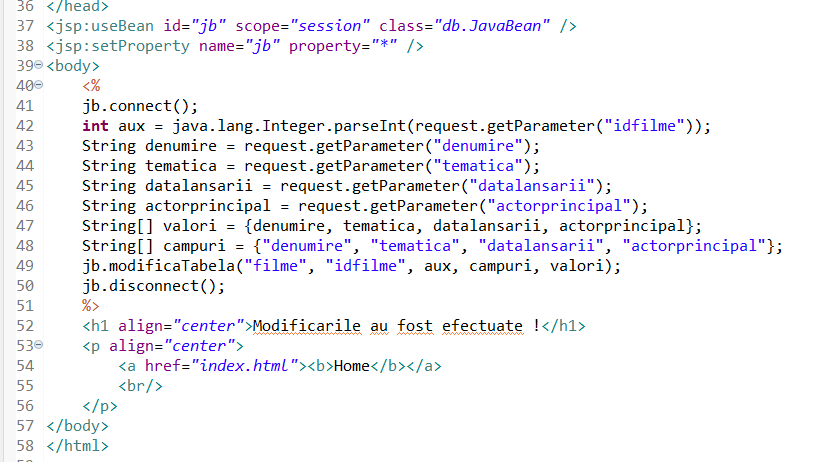
Apasand butonul vom fi redirectionati catre fisierul JSP modifica\_filme.



Se realizeaza un formular in cadrul caruia se ruleaza aceleasi instructiuni ca in cazul operatiei de afisare a datelor. Incepand cu linia 95 cu structura repetitiva while, se plaseaza ‘’cursorul’’ pe prima linie, si se preia din “idfilme” valoarea id-ului ce se leaga de valorile pe care dorim sa le afisam. Singura diferenta este ca formularul are ca si “action” fisierul JSP m1\_filme, ce ia locul unei operatii de adaugare.

In fisierul m1\_filme, am creat campuri de tip INPUT pentru introducerea noilor date. Acestea au fost create in cadrul formularului ce este legat de fisierul m2\_filme.jsp, unde se realizeaza modificarea propriu-zisa.





## 7) Concluzii

Bazele de date sunt prezente in practica peste tot, in special in companii si intreprinderi de productie si comercializare, care le implementeaza in sistemul lor. Pentru o utilizare sigura si corecta, este important sa se creeze o arhitectura care separa functionalitatile, adica partea back-end de partea front-end.

## 8) Bibliografie

1. Cursuri PIBD
2. <https://ro.wikipedia.org/>
3. <https://www.youtube.com/>
4. <https://openai.com/blog/chatgpt/>