**Programarea interfețelor**

**pentru baze de date**

**Documentatie Tehnologie Javafx**

**Cerinta**:Creaţi două aplicaţii care să conţină o bază de date creată în sistemul de gestiune bazelor de date MySql şi două interfeţe la aceasta (baza de date este comună). La crearea interfeţelor se vor folosi două tehnologii(la alegere - ex.: PHP, JSP, Hibernate, JPA, .NET

etc.).

Tehnologia folosita pentru creare a bazei de date:**MySql**. „MySQL este un sistem de gestiune a bazelor de date relaționale, produs de compania suedeza MySQL AB și distribuit sub Licența Publică Generală GNU. Este cel mai popular SGBD open-source la ora actuală, fiind o component cheie a stivei LAMP (Linux, Apache, MySQL, PHP)”[1]

Am realizat o baza de date in MySQL WORKBENCH ,eacontinand 2 tabele : **filme** si **actori**. Asocierea dintre eleeste detipul M:N , ceea ce presupune crearea unei tabele intermediare , pe care am numit-o **roluri.**

Tabela **filme** contine urmatoare coloane:

* idFilm , de tip INT , cheieprimara
* Titlu , de tip VARCHAR
* TipFilm, de tip VARCHAR

Tabela **actori** contine urmatoarele coloane :

* idActor, de tip INT,cheieprimara
* Nume, de tip VARCHAR
* Prenume, de tip VARCHAR
* Nationalitate, de tip VARCHAR

Tabela intermediara **roluri** contine urmatoarele coloane :

* idRol, de tip INT,cheieprimavara
* idActor, de tip INT
* idFilm, de tip INT
* NumePersonaj, de tip VARCHAR

Datorita asocierii M:N , idActor si idFilm sunt chei straine (FK) pentru tabela **roluri.** Cheile primare(PK) corespunzatoare fiecarei tabele au fost setate cu urmatoarele proprietati : not null (NN), auto-increment (AI).

**Cheie primara**:Una sau mai multe coloane ale caror valori identifica in mod unic toate liniile unui table. [2]

Diagrama asociata tabelelor este reprezentata in figura de mai jos :

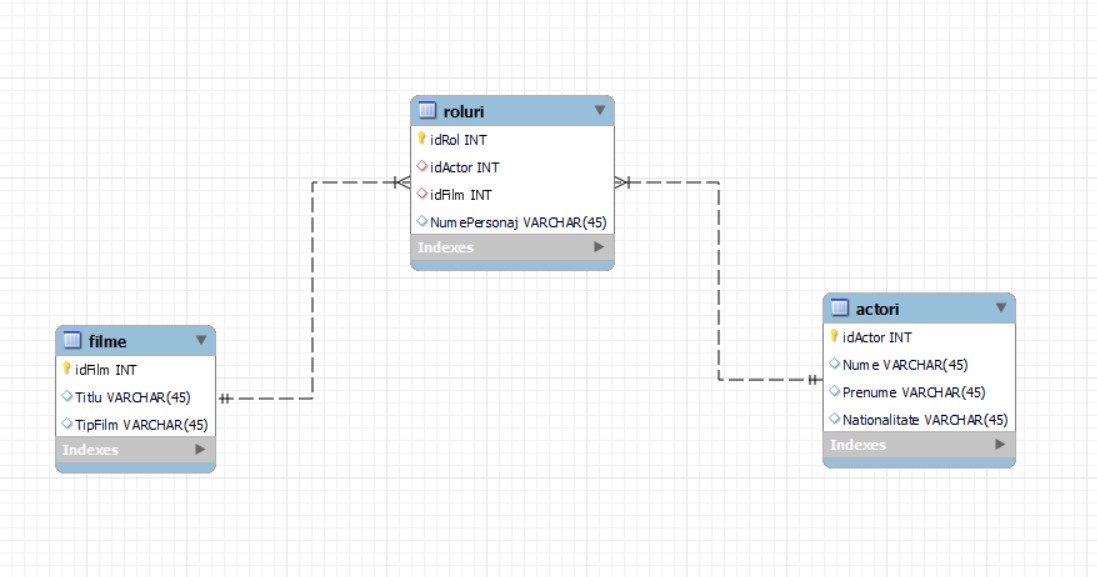


Fig. 1: diagram asociata tabelelor

Pe diagram astfel generată (EER) putem identifica următoarele aspecte:

* Între tabelele filme și roluri– asociere de tipul 1:N
* Între tabelele filme și actori– asociere de tipul M:N
* Între tabelele roluri si actori – asociere de tipul 1:N

**Tehnologia JavaFX**

JavaFX este o platformă/librărie care permite construirea de:

• Aplicaţii desktop (Desktop Applications);

• Aplicaţii internet bogate (Rich Internet Applications - RIAs).

Aplicaţiile dezvoltate cu ajutorul acestei tehnologii pot rula pe

• Sisteme de calcul uzuale;

• Telefoane mobile;

• Tablete;

• Televizoare etc.

Pentru baza mea de date a trebuit sa creez cate o clasa pentru fiecare tabelă:

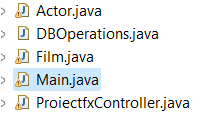


Figura 2. Clasele create

Fiecare clasă contine constructori aferent fiecarui atribut al tabelelor:

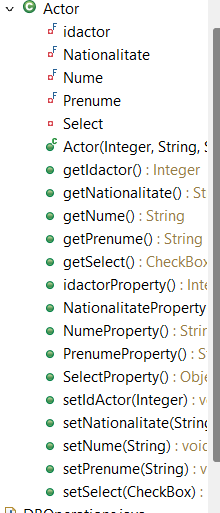
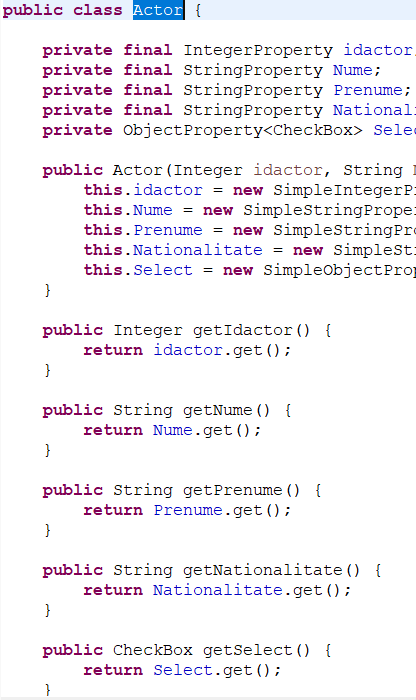
 

Figura3.Setteri&Getteri Figura4.Cod setteri&getteri

In această interfata putem doar să vizualizam datele si sa adaugam.

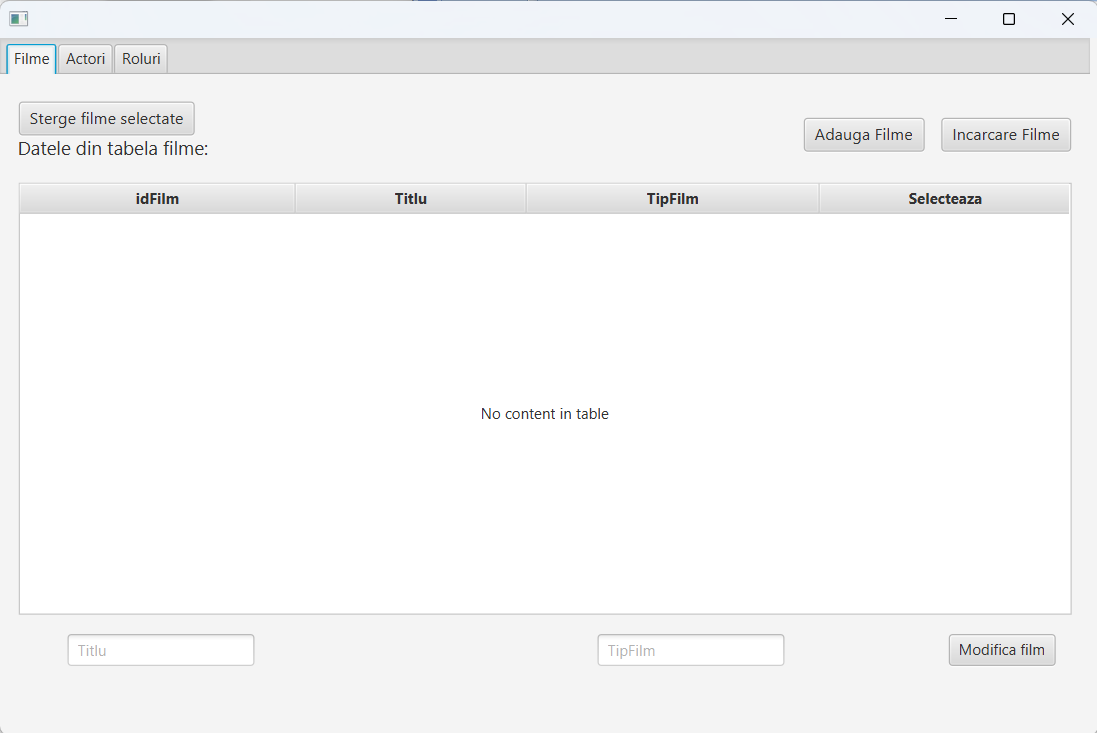
Aceasta este fereastra principala cand rulam aplicatia, construită in Scene Builder:

Figura5.FereastraPrincipala

După apasarea butonului “incarcare Filme/Actori/Roluri”, vor aparea datele existente in baza de date:

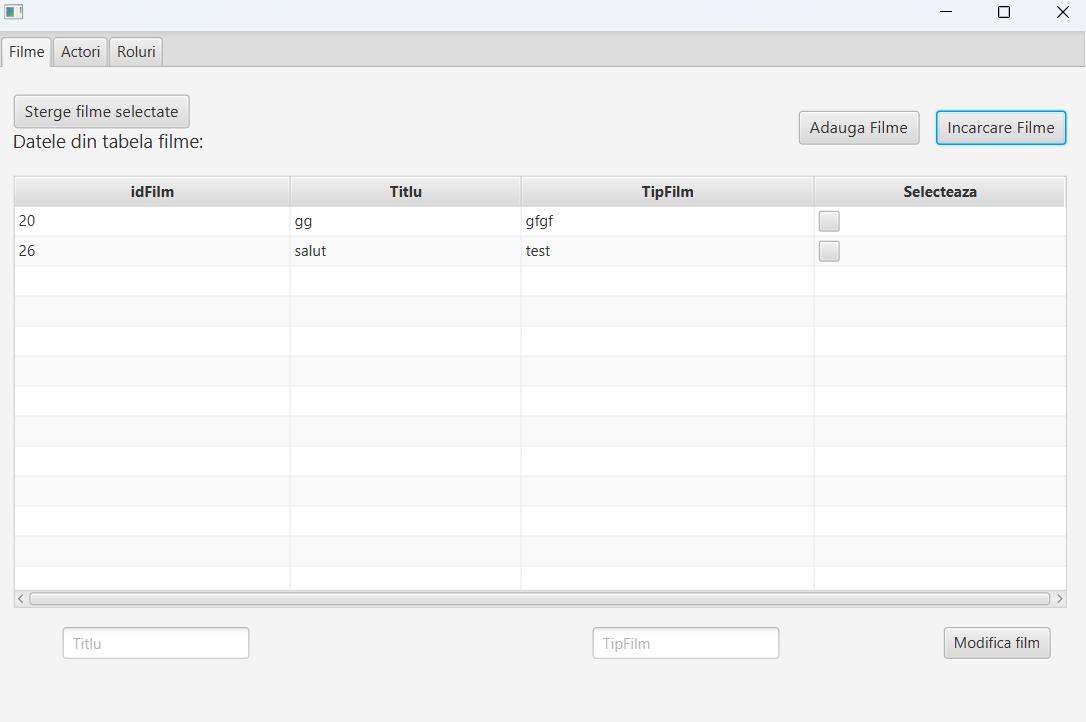


Figura6.Tabela Filme

Fiecare element din această tabelă este initializat in clasa “ProiectFxController”.

Codul pentru functia incarca Filme:

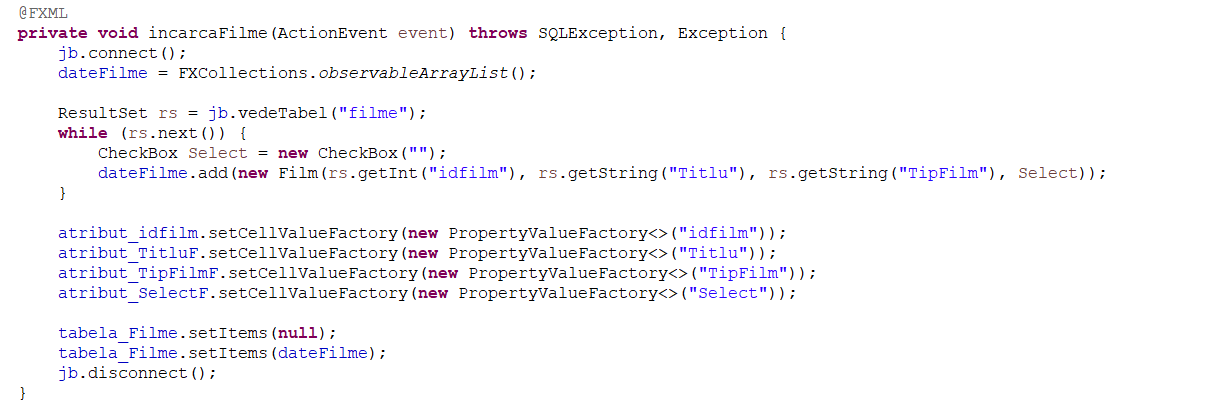


Figura7.Functia incarca Filme

Dupa cum se poate observa, se face conexiunea la baza de date, apoi se adaugă din baza de date filmele cu atributele idfilm, Titlu, TipFilm. După încheierea conexiunii, se setează în tabela Filme elementele continute de dateFilme.

In mod asemanator sunt încarcate si datele din tabela Roluri, doar ca trebuie preluate si atributele idFilm si idActor:



Figura 8. Functia incarcaRoluri

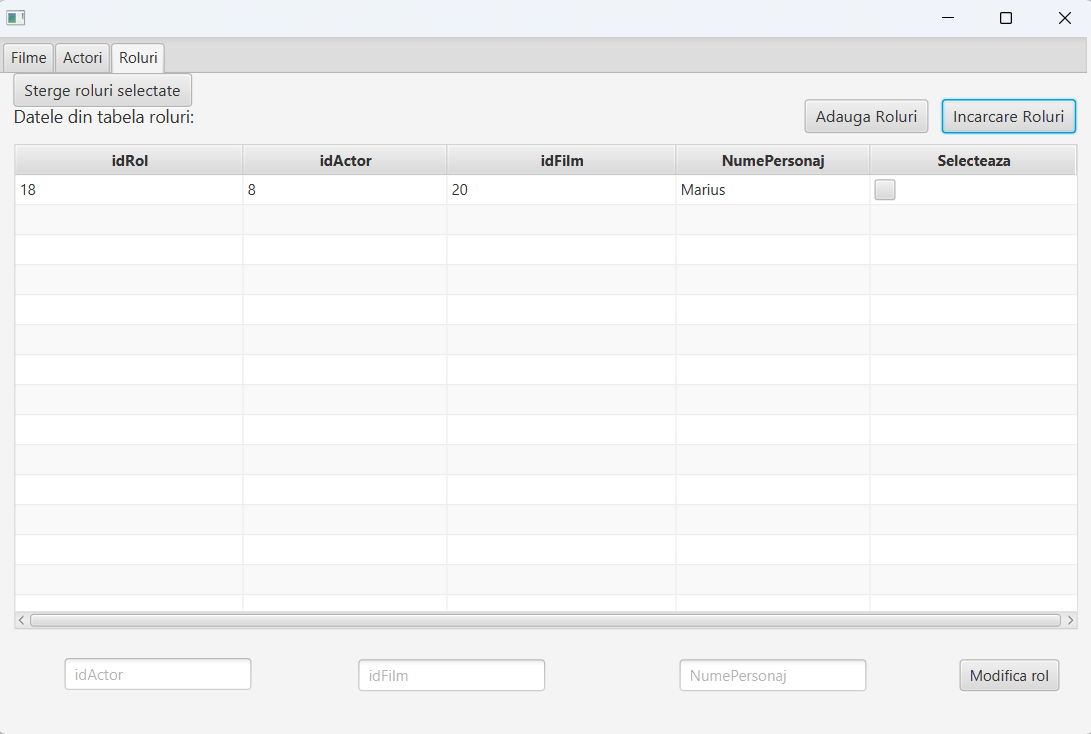


Figura9.Tabela Roluri

Pentru adaugarea unui nou element, trebuie să apasam pe butonul “Adaugă film/actor/rol”.

Pentru exemplu, vom adauga un nou film. Prin apasarea butonului “Adauga film” apelamfunctia “startAdaugaFilm()”:

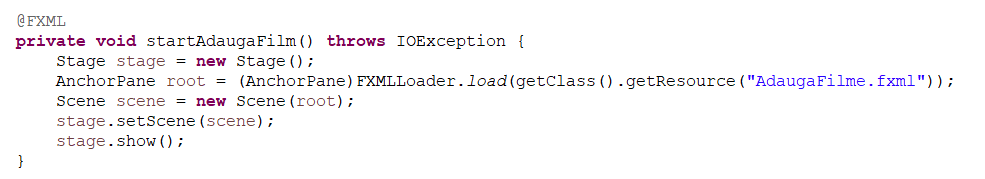


Figura.10 Functia startAdaugaFilm()

Aceasta deschide fereastra “AdaugaFilm :

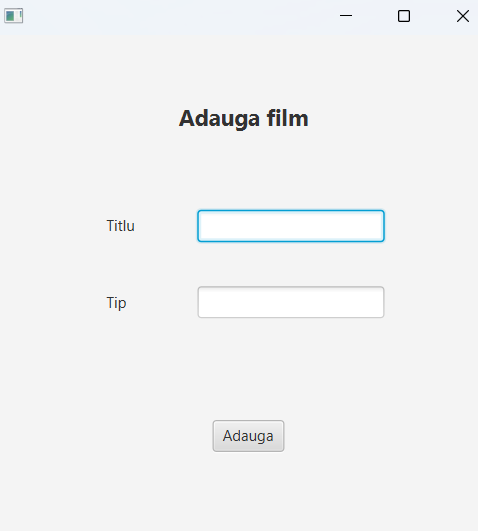


Figura 11. Fereastra Adauga film

Codul corespunzator functiei :

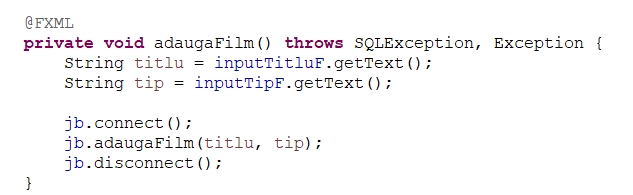


Figura 12. Functia AdaugaFilm()

Instructiunea jb.adaugaFilm(titlu, tip); apeleaza functia prin care se face inserarea in baza de date. Aceasta instructiune se afla in DBOperations.

Functia de adaugare a filmului:

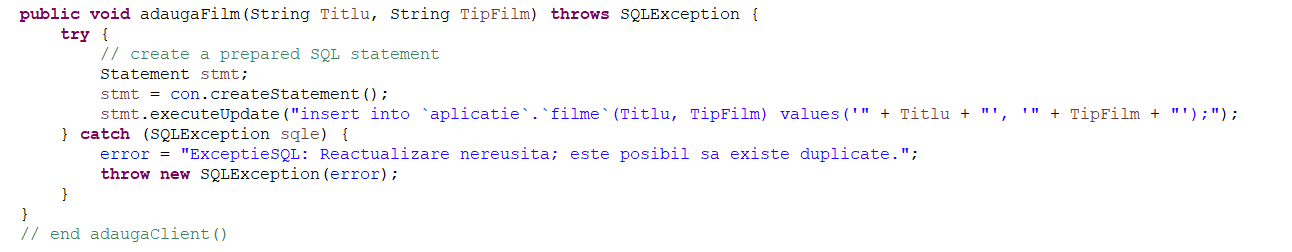


Figura 13. Functia adaugaFilm

Aceasta functie rescrie linia de comanda de inserare din MySQL.

După ce utilizatorul scrie un Titlu si un TipFilm, acesta va apasa pe butonul “Adauga” apoi va apasa din nou pe butonul “Incarcare Filme” pentru a vedea noua linie adaugata .

In cazul tabelei Actori, adaugarea este identica. Pentru adaugarea in tabela roluri, avem ceva mai deosebit, deoarece putem alege, după id, ce film si ce actor dorim sa adaugăm.

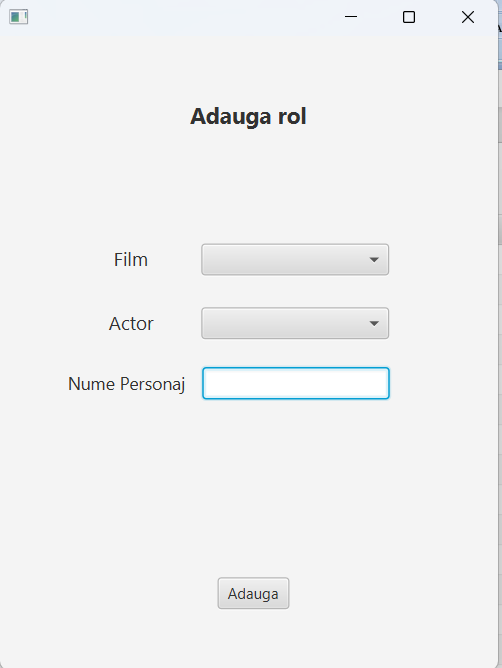


Figura 14. Pagina “Adauga rol”

In această fereastă avem 2 “combobox” pentru filme si actori, iar in rest o caseta de input.

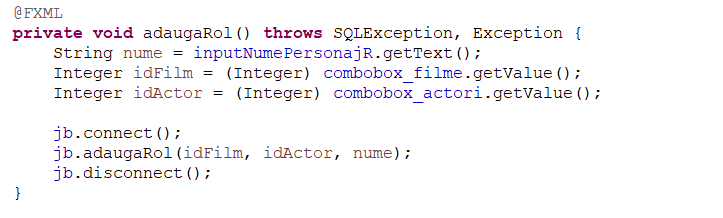


Figura 15. Functia adauagaRol

In aceasta functie sunt preluate datele introduse de la tastatura si id-urile selectate la Film/Actori. Apoi prin apelarea Jb.adaugaRol acestea sunt inserate in tabela Roluri din baza de date.

Pentru cele doua combobox codul aferent este:



Fig 16&17 Functiile pentru “combobox”

Fiecare preia din tabelele Film si Actori toate id urile existente pentru a le pune la alegerea utilizatorului:

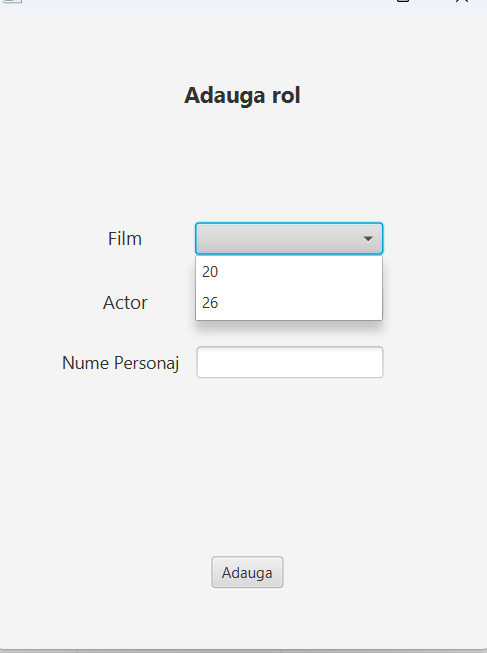


Figura 18. Lista de id-uri a Filmelor

După ce introducem toate datele necesare adaugarii, utilizatorul trebuie să apese pe butonul “Adaugă”, sa inchida fereastra si sa apese din nou pe butonul “Incarcare Roluri” pentru a reincarca tabela.

În cazul ștergerii unui film, va trebui selectat în tabelul rândul corespunzător intrării din baza de date ce se dorește a fi șters, apoi apăsat butonul “Sterge Film selectat”.

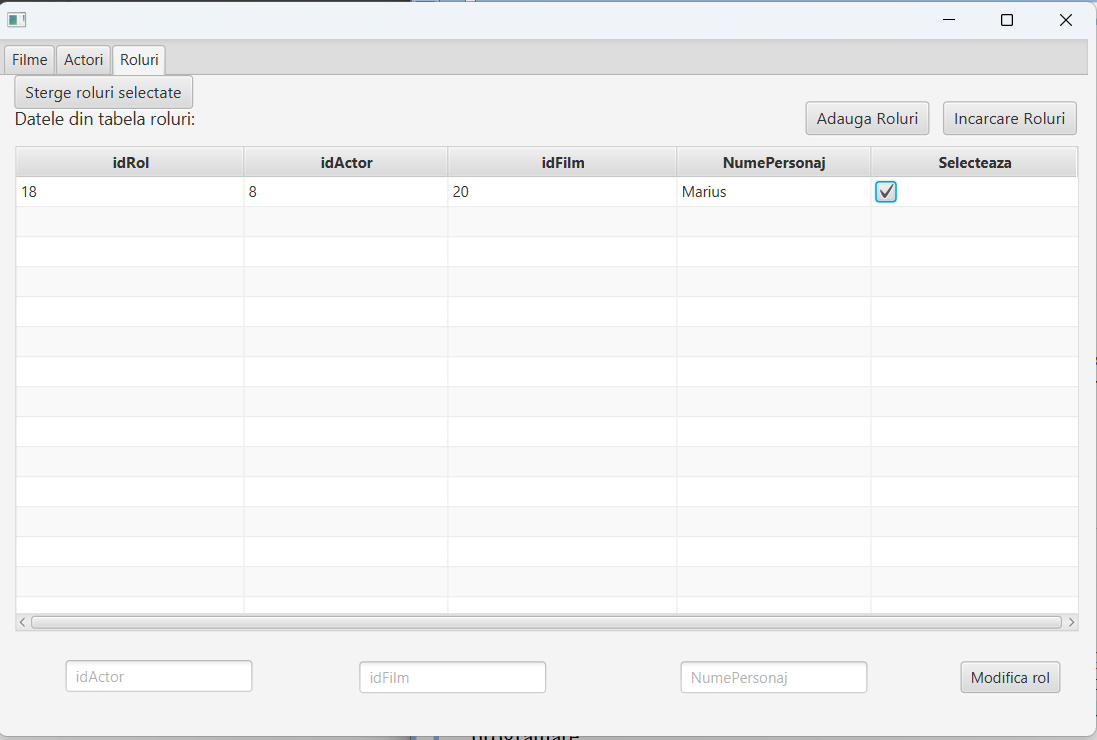


Figura 19. Stergerea unui film

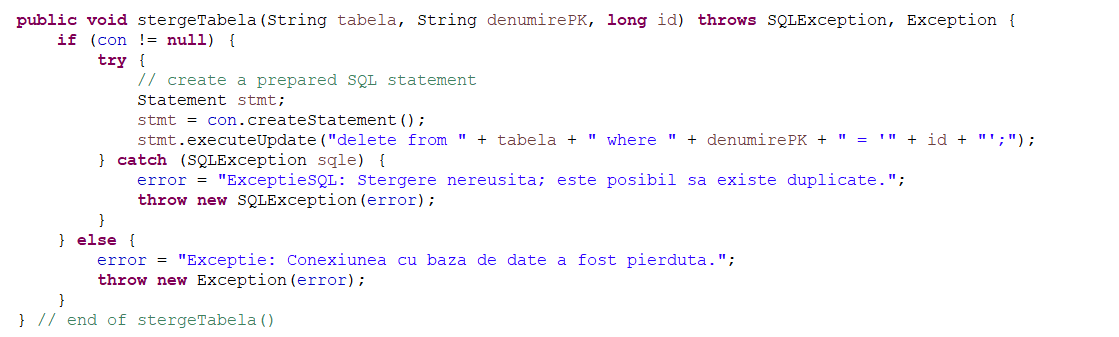


Figura 20: DBOperations.java code pentru stergere

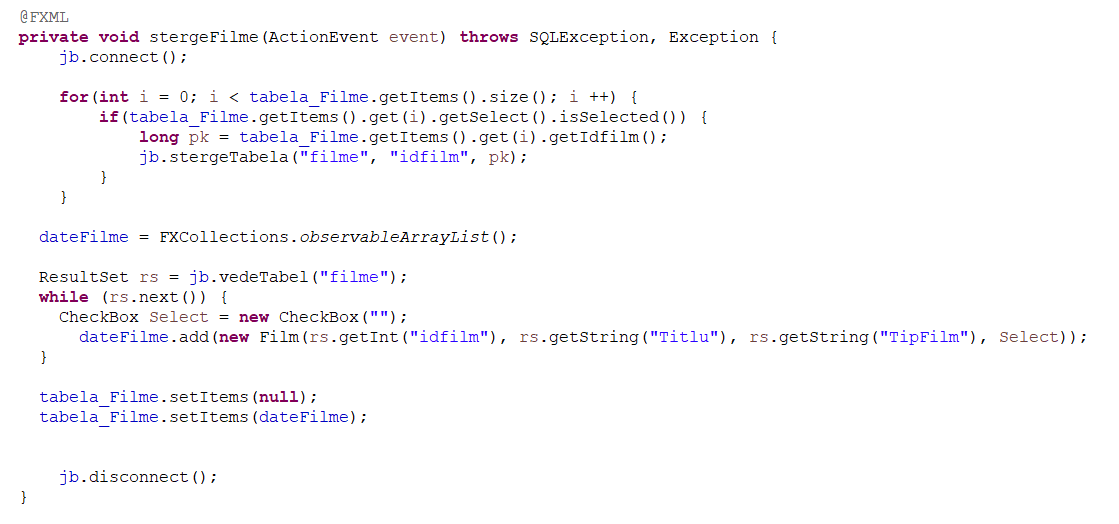


Figura 21: ApplicationController.java code pentru stergere

Similar se va proceda si pentru restul tabelelor.

În cazul modificării unui actor, va trebui selectat în tabelul rândul corespunzător

intrării din baza de date ce se dorește a fi modificat, apoi apăsat butonul “Modifica actor”.

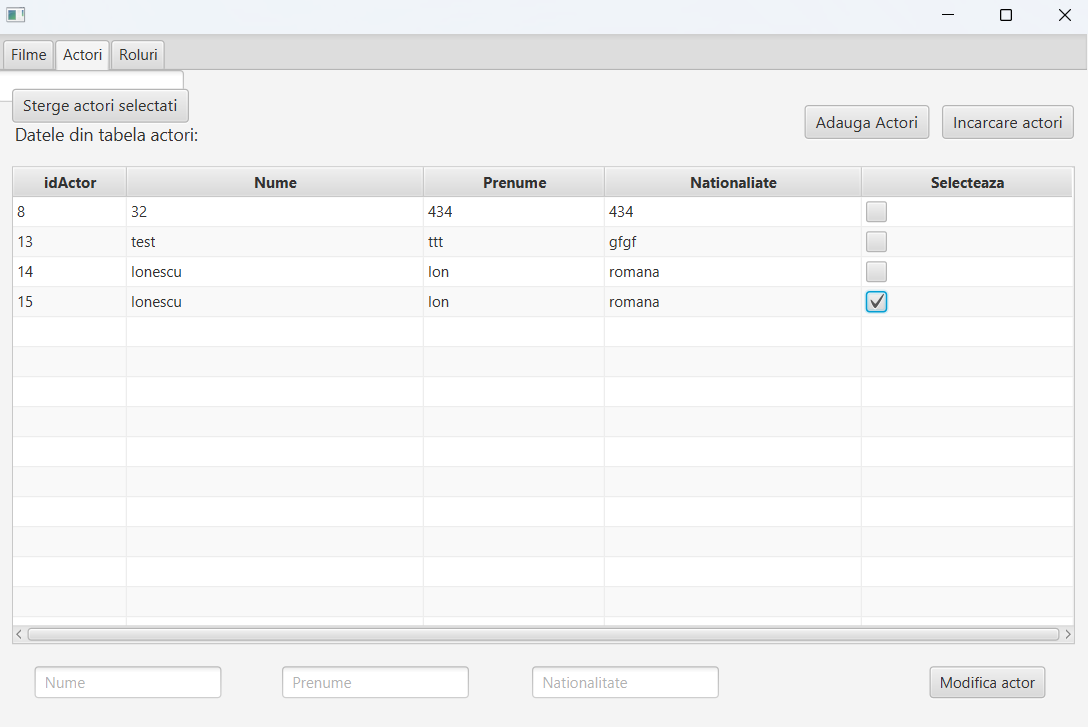


Figura 22: Modificare date actori

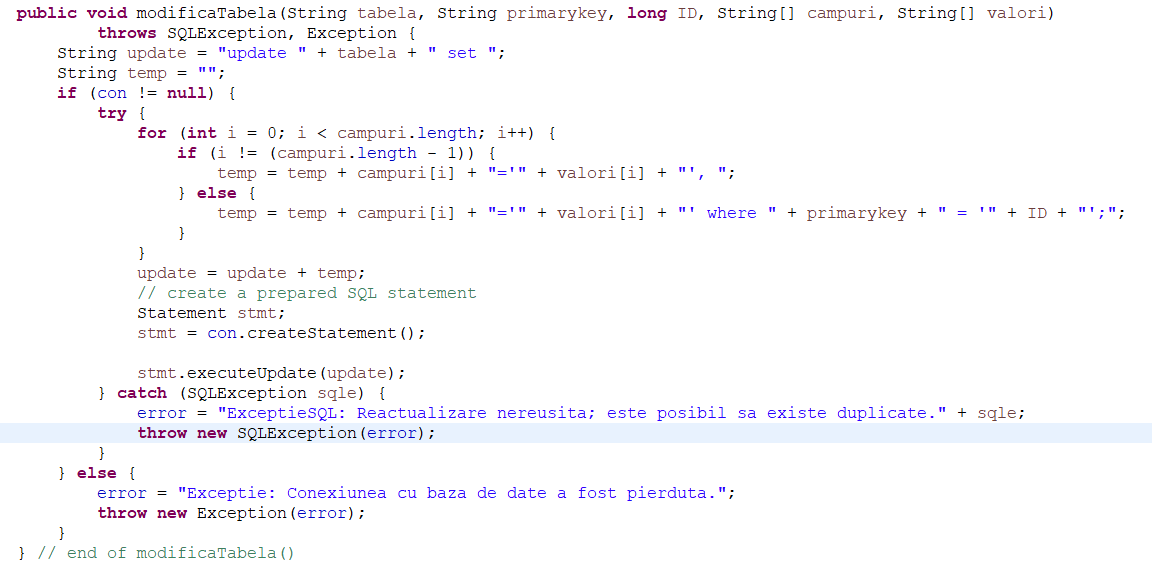


Figura 23: DBOperations.java code pentru modificare tabela

@FXML

**private** **void** modificaActori(ActionEvent event) **throws** SQLException, Exception {

jb.connect();

ArrayList<String> campuri = **new** ArrayList<>();

ArrayList<String> valori = **new** ArrayList<>();

String nume = inputNumeA.getText();

**if**(!nume.isEmpty()) {

valori.add(nume);

campuri.add("nume");

}

String prenume = inputPrenumeA.getText();

**if**(!prenume.isEmpty()) {

valori.add(prenume);

campuri.add("prenume");

}

String nationalitate = inputNationalitateA.getText();

**if**(!nationalitate.isEmpty()) {

valori.add(nationalitate);

campuri.add("nationalitate");

}

String[] campuri\_array = campuri.toArray(**new** String[0]);

String[] valori\_array = valori.toArray(**new** String[0]);

**for**(**int** i = 0; i < tabela\_Actori.getItems().size(); i ++) {

**if**(tabela\_Actori.getItems().get(i).getSelect().isSelected()) {

**long** pk = tabela\_Actori.getItems().get(i).getIdactor();

jb.modificaTabela("actori", "idActor", pk, campuri\_array, valori\_array);

}

}

dateActori = FXCollections.*observableArrayList*();

ResultSet rs = jb.vedeTabel("actori");

**while** (rs.next()) {

CheckBox Select = **new** CheckBox(**null**);

dateActori.add(**new** Actor(rs.getInt("idactor"), rs.getString("Nume"), rs.getString("Prenume"), rs.getString("Nationalitate"), Select));

}

inputNumeA.clear();

inputPrenumeA.clear();

inputNationalitateA.clear();

tabela\_Actori.setItems(**null**);

tabela\_Actori.setItems(dateActori);

jb.disconnect();

}

Figura 24: ApplicationController.java code pentru modificare actori

Concluzii

In concluzie, din acest curs am invatat sa realizam o interfata ce permite gestionarea unor date dintr-o baza de date MySQL chiar si de catrepersoane care nu au cunostinte legate de SQL sau de limbaje de programare.

Folosind interfata creata, o firma de casting pentru filme isi poate gestiona afacerea, adaugand noi actori, noi filme si roluri noi. De asemenea datorita tehnologiilor Javafx si MySQL aplicatia poate lucra pe un server putand fi accesata din mai multe locatii de pe tot globul, fiind necesara doar o conexiune de internet.

Practic in acest mod nu mai este nevoie ca un Database Admin sa administreze baze de date a firmei respective intr-un program de management a bazelor de date precum PhpMyAdmin.

# Bibliography

|  |  |
| --- | --- |
| [1] | Wikipedia, "MySQL," [Online]. Available: https://ro.wikipedia.org/wiki/MySQL. |
| [2] | D. Avasilica, "Introducere in SQL," [Online]. Available: http://etutoriale.ro/articles/100/1/Introducere-in-SQL/. |

Alte surse de inspiratie pentru proiect:

* Curs de Programare a Interfetelor pentru Baze de Date (V. Pupezescu)