Documentatie Proiect la

Programarea Interfetelor

pentru Baze de Date

- Tehnologie JavaFX -

Dumitru Ana-Maria

Grupa:434A

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 primita, am ales 2 tehnologii diferite: JSP si JavaFX. In acest document voi prezenta ce am realizat pentru technologia JSP.

Tehnologia folosita pentru baza de date este: 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 componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP)” [1]

Am realizat o baza de date in MySQL Workbench, ea avand 3 tabele: elevi, teme si realizare. Realizare reprezinta tabela de legatura intre elevi si teme, existand o relatie M:N intre ele.

Tabela elevi contine urmatoarele coloane:

* idelevi , de tip INT , cheie primara
* nume , de tip VARCHAR
* prenume, de tip VARCHAR
* clasa , de tip VARCHAR
* medie, de tip INT

Tabela teme contine urmatoarele coloane:

* idelevi , de tip INT , cheie primara
* materie, de tip VARCHAR
* subiect , de tip VARCHAR
* punctaj, de tip INT

Tabela realizare contine urmatoarele coloane:

* idrealizare , de tip INT , cheie primara
* idelevi , de tip INT , cheie primara
* idteme , de tip INT , cheie primara
* termen , de tip VARCHAR
* tip, de tip VARCHAR
* format , de tip VARCHAR

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

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

A screen shot of a computer

Description automatically generated

**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:

A screenshot of a computer program

Description automatically generated

Figura2: Clase create

Fiecare clasă contine constructori aferenti fiecarui atribut al tabelelor: A screenshot of a computer program

Description automatically generated

Figura3.Setteri&Getteri

A screenshot of a computer program

Description automatically generated

 Figura4.Cod setteri&getteri

In această interfata putem să vizualizam datele, sa adaugam, sa modificam si sa stergem.

A screenshot of a computer

Description automatically generated

Figura 5.Fereastra Principala

După apasarea butonului “incarcare Elevi/Teme/Realizare”, vor aparea datele existente in baza de date:

A screenshot of a computer

Description automatically generated

   Figura 6.Tabela Elevi

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

Codul pentru functia incarcaElevi:

A close-up of a computer screen

Description automatically generated

Dupa cum se poate observa, se face conexiunea la baza de date, apoi se adaugă din baza de date produsele cu atributele idelevi, nume, prenume, clasa, media. După încheierea conexiunii, se setează în tabela\_Elevi elementele continute de dateElevi.

In mod asemanator sunt încarcate si datele din tabela Realizare, doar ca trebuie preluate si atributele idelevi si idteme:

A screen shot of a computer code

Description automatically generated

Figura 8. Functia incarcaRealizare

A screenshot of a computer

Description automatically generated

Figura 9: Tabela Realizare

Pentru adaugarea unui nou element, trebuie să apasam pe butonul “Adaugă elev/tema/realizare”.

Pentru exemplu, vom adauga un nou elev. Prin apasarea butonului “Adauga un nou elev” apelam functia “startAdaugaElevi()”:

A computer screen shot of a loading program

Description automatically generated

Aceasta deschide fereastra “AdaugaElevi”

A screenshot of a computer

Description automatically generated

Codul corespunzator functiei :

A screen shot of a computer code

Description automatically generated

Figura 11. Functia AdaugaElevi

Instructiunea jb.adaugaElev(nume,prenume,clasa,medie); apeleaza functia prin care se face inserarea in baza de date.

Aceasta instructiune se afla in DBOperations. Functia de adaugare a produsului:

A screenshot of a computer

Description automatically generated

Figura 13. Functia adaugaElev

Aceasta functie rescrie linia de comanda de inserare din MySQL.

După ce utilizatorul scrie un Nume, Prenume,Clasa si Medie, acesta va apasa pe butonul “Adauga” apoi va apasa din nou pe butonul “Incarcare Elevi” pentru a vedea noua linie adaugata .

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Fig 14&15 Adaugarea unui elev

Putem identifica noul produs avand idprodus=23.

In cazul tabelei Teme, adaugarea este identica. Pentru adaugarea in tabela Realizare, avem ceva mai deosebit, deoarece putem alege, după id, ce elev si ce tema dorim sa adaugăm.

A screenshot of a computer

Description automatically generated

Fig 16

In această fereastă avem 2 “combobox” pentru Elevi si Teme iar in rest 3 casete de input.

A computer screen shot of a program

Description automatically generated

Fig 17 functia Adauga Realizare

In aceasta functie sunt preluate datele introduse de la tastatura si id-urile selectate la Elevi/Teme. Apoi prin apelarea  Jb.adaugaRealizare acestea sunt inserate in tabela Realizare din baza de date.

A screenshot of a computer

Description automatically generated

Fig 18 Functia adaugaRealizare

Pentru  cele doua combobox codul aferent este:

**public** **void** initialize(URL url, ResourceBundle rb) {

jb = **new** DBOperations();

jb = **new** DBOperations();

**if**(combobox\_elevi != **null**) {

ObservableList<Integer>dateidElevi = FXCollections.*observableArrayList*();

**try** {

jb.connect();

ResultSet rs = jb.vedeTabela("elevi");

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

dateidElevi.add(rs.getInt("idelevi"));

}

jb.disconnect();

combobox\_elevi.setItems(dateidElevi);

}**catch**(Exception e) {

System.***err***.println(e.getMessage());

}

}

**if**(combobox\_teme != **null**) {

ObservableList<Integer>dateidTeme = FXCollections.*observableArrayList*();

**try** {

jb.connect();

ResultSet rs = jb.vedeTabela("teme");

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

dateidTeme.add(rs.getInt("idteme"));

}

jb.disconnect();

combobox\_teme.setItems(dateidTeme);

}**catch**(Exception e) {

System.***err***.println(e.getMessage());

}

}

**if**(combobox\_eleviM != **null**) {

ObservableList<Integer>dateidElevi = FXCollections.*observableArrayList*();

**try** {

jb.connect();

ResultSet rs = jb.vedeTabela("elevi");

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

dateidElevi.add(rs.getInt("idelevi"));

}

jb.disconnect();

combobox\_eleviM.setItems(dateidElevi);

}**catch**(Exception e) {

System.***err***.println(e.getMessage());

}

}

**if**(combobox\_temeM != **null**) {

ObservableList<Integer>dateidTeme = FXCollections.*observableArrayList*();

**try** {

jb.connect();

ResultSet rs = jb.vedeTabela("teme");

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

dateidTeme.add(rs.getInt("idteme"));

}

jb.disconnect();

combobox\_temeM.setItems(dateidTeme);

}**catch**(Exception e) {

System.***err***.println(e.getMessage());

}

}

}

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

A screenshot of a computer

Description automatically generated

Figura 18. Lista de id-uri a Elevilor

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 Realizare” pentru a reincarca tabela.

A screenshot of a computer

Description automatically generated

Figura 19. Tabela Realizare dupa adaugarea unei noi Realizari

Pentru functionalitatea de stergere vom selecta cu mouse-ul linia pe care o dorim sa o stergem si apoi vom apasa pe butonul de stergere. Se poate cate un element pe rand.

A screenshot of a computer

Description automatically generated

Figura 20. Inainte de stergere

A screenshot of a computer

Description automatically generated

Figura 21. Dupa stergere

Functia de stergere este implementata in controller. Se obtine id-ul si indexul elevului ce se doreste a fi sters si apoi se utilizeaza functia stergeDateTabela din DBOperations.

A computer screen shot of a computer code

Description automatically generated

Figura 22.

Functia stergeTeme

A screenshot of a computer

Description automatically generated

Figura 23. Functia stergeDateTabela

Stergerea se bazeaza pe acelasi principiu pentru toate cele 3 tabele.

Pentru modificarea unui element, trebuie să apasam pe butonul “Modifica elev/tema/realizare”.

Pentru exemplu, vom adauga un nou elev. Prin apasarea butonului “Adauga un nou elev” apelam functia “startModificaElevi()”:

A blue screen with white text

Description automatically generated

Figura 24. Functia startModificaElevi

Index\_of\_modified\_object este variabila ce retine index-ul obiectul pe care il dorim sa il stergem.

Aceasta deschide fereastra “ModificaElevi”

A screenshot of a computer

Description automatically generated

Figura 25. Fereastra ModificaElevi

Codul corespunzator functiei :

A computer screen shot of a code

Description automatically generated

Figura 26. Functia AdaugaElevi

Instructiunea jb.modificaTabela("teme", "idteme", *index\_of\_modified\_object*, campuri, valori); apeleaza functia prin care se face inserarea in baza de date.

Aceasta instructiune se afla in DBOperations. Functia de adaugare a produsului:

A screenshot of a computer

Description automatically generated

Figura 27. Functia modificaTabela

Aceasta functie rescrie linia de comanda de inserare din MySQL.

După ce utilizatorul scrie un Nume, Prenume,Clasa si Medie, acesta va apasa pe butonul “Modifica” apoi va apasa din nou pe butonul “Incarcare Elevi” pentru a vedea linia modificata .

A screenshot of a computer

Description automatically generated

Figura 28.Fereastra de modificare

A screenshot of a computer

Description automatically generated

Fig 29 Modificare unui elev

Putem identifica produsul modificat avand idprodus=22.

Modificarea arata si este implementata asemenator cu adaugarea unui element.

Concluzii

In concluzie acest curs ne-a aratat  ca putem face destul de usor o interfata prietenoasa ce permite gestionarea unor date dintr-o baza de date MySQL chiar si de catre persoane care nu au cunostiinte legate de SQL sau de limbaje de programare.

Fisierele de tip FXML sunt destul de usor de create, cu ajutorul aplicatiei SceneBuilder.

Folosind interfata creata, o firma sau un depozit de exemplu isi poate gestiona marfa si furnizorii, adaugand noi produse, noi componente si furnizori noi. Nu le trebuie un angajat special pentru gestionarea datelor, orice angajat existent poate face acest lucru folosind aceasta interfata. De asemenea datorita tehnologiilor JavaFX si MySQL aplicatia poate lucra pe un server putand fi accesata din mai multe locatii, fiind necesara doar o conexiune de internet.

Bibliografie

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