A close-up of a white background

Description automatically generated

**PROGRAMAREA INTERFEȚELOR PENTRU BAZE DE DATE**

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

Student: **Radu Crina**

Grupa: **433B**

**Cuprins**

1. Tema Proiectului
2. Descrierea sistemului de gestiune a bazelor de date MySQL
3. Limbajul HTML si utilitatea sa in aplicatii
4. Tehnologia Hibernate utilizata
5. Descrierea aplicației
6. Concluzii
7. Bibliografie
8. **Tema Proiectului**

Tema proiectului se bazează pe dezvoltarea unei aplicații ce conține o bază de date, creată în sistemul de gestionare a bazelor de date MySQL. Se pot utiliza diferite tehnologii: JSP, Hibernate, JPA. In acest proiect am utilizat tehnologia Hibernate. Interfețele vor trebui să permită utilizatorului să execute urmatoarele operații pe toate tabele: vizualizare, adaugare, modificare și ștergere de date. Vizualizarea tabelelor de legatură va presupune vizualizarea datelor referite din celelalte tabele. Asocierea pentru tabelele din baza de date este de M:N.

1. **Descrierea sistemului de gestiune a bazelor de date MySQL**

MySQL este un sistem de gestionare a bazelor de date relaționale open source care este utilizat în principal pentru aplicațiile online. MySQL poate crea și gestiona baze de date foarte utile. Acesta permite stocarea, organizarea și recuperarea informațiilor într-o structură de tabel cu rânduri și coloane. Este compatibil cu majoritatea sistemelor de operare, inclusiv Windows, Linux și macOS.

Pentru a administra bazele de date MySQL se poate folosi modul linie de comanda sau, prin descărcare, de pe internet, o interfață grafică: MySQL Administrator și MySQL Query Browser. Baza de date MySQL este folosită în principal ca mijloc de stocare de date pentru aplicații mari, bazate pe web.

1. **Limbajul HTML si utilitatea sa in aplicatii**

HTML (HyperText Markup Language) este un limbaj de marcare utilizat pentru crearea de pagini web. Folosind etichete HTML, poți defini structura și conținutul unei pagini web, precum titluri, paragrafe, link-uri, imagini și multe altele. Rezultatul final este o pagină web care poate fi vizualizată în browser-ul web. HTML este esențial pentru crearea de site-uri web și este unul dintre elementele de bază ale tehnologiei web.

1. **Tehnologia Hibernate utilizata**

Tehnologia Hibernate este un instrument in cadru relational de obiecte sau cadu pentru limbajul de programare Java. Este software-ul gratuit sub GNU Lesser General Public License 2.1, si isi propune sa ofere utilizatorului cardul pentru a gestiona maparea impedantelor relationale obiecte. Arhitectura tehnologiei Hibernate cuprinde:

* Un cod de aplicatie Java ce consta in toate clasele, variabilele si obiectele care definesc logica de afaceri a aplicatiei. Aceasta clasa comunica cu Hibernate.
* Utilizarea API-ului de baza Java, Java Database Connectivity(JDBC), Java Transaction API (JTA), Java Naming si Directory Interface (JNDI) pentru a comunica cu baza de date cu scopul de a afla starea unui obiect executand citirea, crearea, actualizarea, stergerea (operatiile CRUD).

1. **Descrierea aplicației**

* **Baza de date**

Tema se bazează pe crearea unei baze de date ce are doua tabele în asociere M:N. Tabelele sunt: clienti și comenzi. Pentru cele doua tabele am ales câteva atribute caracteristice:

Pentru tabela clienti am ales ca și cheie primara idclient. Celelalte atribute sunt nume, adresa, telefon, mail.

A screenshot of a computer

Description automatically generated

***Figura 1.*** *Tabela client*

Pentru tabela comenzi am ales ca și cheie primara idcomanda. Celelalte atribute sunt valoare, tipprodus, cantitate.

A screenshot of a computer

Description automatically generated

***Figura 2****. Tabela comenzi*

Asocierea M:N (Many-to-Many) este o relație de bază de date între două tabele care permite fiecărei înregistrări din prima tabelă să fie asociate cu mai multe înregistrări din a doua tabelă și invers. Această relație necesită crearea unei tabele intermediare pentru a gestiona această asociere. În cazul nostru, am creat tabela de legătura furnizori. Am ales ca și cheie primara idfurnizor. Celelalte atribute sunt idclient, idcomanda, adresa, tipprodus. În această nouă tabelă, atributele ce au fost selectate ca și chei primare pentru tabelele anterioare, vor deveni chei străine (FK) pentru tabela de legătură.

A screenshot of a computer

Description automatically generated

***Figura 3.*** *Tabela furnizori*

* **Diagrama logică a bazei de date**

*A screenshot of a computer

Description automatically generated*

***Figura 4.*** *Diagrama logică a bazei de date*

Această diagramă oferă o reprezentare logică și detaliată a datelor din cele 3 tabele.

Între clienti și furnizori există o asociere de tip 1:N. Între clienti si comenzi exista o asociere de tip M:N. Între comenzi și furnizori există o asociere de tip 1:N.

* **Functionalitatea aplicatiei**

Intalnim partea de model:

* Identificata in POJO, unde se face maparea. In clasele din acest package, cele mai importante elemente sunt functiile GET si SET. Cand se fac operatii pe obiectele din clasa, modificarile se “resimt” in baza de date

A screenshot of a computer

Description automatically generated

***Figura 5.*** *POJO Package*

Partea de functionalitate si de vizualizare:

Intalnim urmatoarele pachete: Controller, DAO, DaoImpl. Aici se fac operatii de baza pe obiectele din clase, aceasta parte fiind partea dinamica a proiectului. In DAO avem interfete pentru standardizarea denumirilor ce au fost implementate in DAOImpl, unde toate operatiile de aici sunt facute in cadrul unor tranzactii. Controller-ul este implementat sub forma de SERVLET.

A screenshot of a computer program

Description automatically generated

***Figura 6.*** *Controller, DAO, DAOimpl Package*

* Vizualizarea este implementata prin JSP, prin limbajul HTML.

A screenshot of a computer

Description automatically generated

***Figura 7****. Webapp*

JSP este o tehnologie care permite dezvoltatorilor sa genereze pagini web dinamice, utilizand fragmente de cod Java, introduse in pagini HTML. Hibernate functioneaza ca un serviciu de mapare (ORM).

In pachetul POJO, luam ca si exemplu clasele Clienti.java si Comenzi.java. Se declara atributele ce vor fi mapate si vor fi, mai apoi, utilizate in cadrul functiilor aplicatiei:

A screen shot of a computer program

Description automatically generated

***Figura 8.*** *Clasa Clienti*

*A screen shot of a computer program

Description automatically generated*

***Figura 9.*** *Clasa Comenzi*

Echivalentul liniilor din tabelele noastre vor fi obiectele instanta la clase.

Avem nevoie de cate o metoda de get si set pentru fiecare atribut din cele doua clase:

*A screen shot of a computer program

Description automatically generated*

***Figura 10.*** *Functiile GET si SET, clasa Clienti*

*A screen shot of a computer program

Description automatically generated*

***Figura 11.*** *Functiile GET si SET, clasa Comenzi*

Constatam prezenta unor fisiere de tip xml pentru functia de configurare. Aici se face maparea propriu-zisa.

A screen shot of a computer program

Description automatically generated

***Figura 12.*** *Clienti.hbm.xml, mapare*

Observam partea de obiect-orientat si de relationare prin intermediul liniilor:

<property name="nume" type="string"> si <column name="nume" length="45" />

In pachetul DAOImpl, clasa HibernateUtil ne ajuta sa obtinem o sesiune pentru a face modificarile in ceea ce priveste datele din baza de date.

*A screen shot of a computer program

Description automatically generated*

***Figura 13.*** *Clasa HibernateUtil*

Fisierul hibernate.cfg.xml este unul de configurare. Putem observa configurarea driverului de MySQL, partea de conetica, prin API, si totodata maparea legata de clasele din POJO.

In fisierul index.html, am implementat functia de afisare pentru datele din tabelele noastre

*A screen shot of a computer program

Description automatically generated*

***Figura 14.*** *index.html*

Observam trei butoane: Afiseaza Comenzi, Afiseaza Clienti, Afiseaza Furnizori. Pentru fiecare buton implementat , avem o caracteristica de “submit” ce ne va face legatura cu functiile din pachetul Controller. De exemplu, pentru afisarea clientilor inregistrate in baza de date, in ClientiController, functia de afisare este implementanta astfel:

*A screen shot of a computer code

Description automatically generated*

***Figura 15.*** *Functia “afiseazaClienti”*

Observam o lista de obiecte de tip clienti. Aceasta lista trebuie sa fie umpluta de obiecte de acest tip, asadar se face legatura cu DAOImpl, unde se creeaza o sesiune, in cadrul careia se face o interogare(“From clienti”). Aceasta este echivalenta cu comanda din MySQL select \* from. Datele vor fi vizualizate in cadrul paginii tabela\_Clienti:

A screenshot of a computer

Description automatically generated

***Figura 16.*** *Tabela Clienti*

Pentru a adauga datele, se creeaza un formular pentru a prelua parametrii si de a-i adauga in baza de date:

A screenshot of a computer program

Description automatically generated

***Figura 17****. Adaugare date*

Cand apasam butonul “submit”, se va face legatura cu servlet-ul ClientiController si prin metoda GET vom obtine parametrii doriti.

A screen shot of a computer code

Description automatically generated

***Figura 18.*** *Functia adaugaClient*

Functia propriu-zisa de adaugare este implementata in ClientiDaoImpl astfel:

A screen shot of a computer screen

Description automatically generated

***Figura 19.*** *Functia adaugaClienti*

Operatia se face intr-o sesiune. Se incepe o tranzactie, datele sunt adaugate in baza de date si apoi tranzactia se inchide.

A screenshot of a computer

Description automatically generated

***Figura 20.*** *Adauga client*

Pentru a sterge datele, se creeaza un nou formular, diferit de cel in care aplicam functia de adaugare. In noul formular, prin metoda POST, facem legatura cu ClientiController prin alegerea a doua functii: modificaClient sau stergeClient.

A screen shot of a computer code

Description automatically generated

***Figura 21.*** *Functiile modificaClient si stergeClient*

Se creeaza din nou o sesiune, se incepe o noua tranzactie. Obiectul este modificat sau sters din baza de date.

Pentru tabela de legatura functiile difera putin deoarece apar si asocieri cu cele doua tabele. In fisierul adauga\_Furnizori.jsp, lista de formatii si lista de membrii sunt incarcate cu functiile din DAOImpl:

A screen shot of a computer code

Description automatically generated

***Figura 22.*** *Asocieri*

A screenshot of a computer

Description automatically generated

***Figura 23.*** *Adauga un nou furnizor*

* **Partea de front-end**

Pentru un aspect mai placut al interfetei, am folosit limbajul HTML combinat cu CSS.

A screenshot of a computer

Description automatically generated

***Figura 24.*** *Aspectul paginii*

1. **Concluzii**

Bazele de date sunt utilizate peste tot, fiecare companie, intreprindere, avand nevoie de o baza de date. Totodata, aplicatia realizata in tehnologia Hibernate si prezentata in cadrul acestui proiect, numita “Sistem de Administrare al Clienților”, permite utilizatorului sa efectueze operatiile cerute pe baza de date creata in MySQL: afisare, adaugare, modificare si stergere.

1. **Bibliografie**

* Cursuri PIBD
* <https://www.nav.ro/blog/ce-este-mysql/>
* <https://ro.wikipedia.org/>
* <https://web.ceiti.md/lesson.php?id=1>