



Universitatea POLITEHNICA din București

Facultatea de electronică, Telecomunicații și Tehnologia Informației



PROGRAMAREA INTERFETELOR PENTRU BAZA DE DATE

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

STUDENT: CANDEA SEBASTIAN CONSTANTIN
GRUPA: 431Aa

CUPRINS

1) Tema proiectului.....	pag. 3
2) Descrierea sistemului de gestiune a bazelor de date MySQL.....	pag.3
3) Tehnologia JSP utilizata in dezvoltarea aplicatiei.....	pag.4
4) Limbajul HTML si utilitatea sa in aplicatii.....	pag. 4
5) Descrierea Aplicatiei.....	pag. 4
6) Concluzii.....	pag.10
7) Bibliografie.....	pag.10

1) Tema proiectului

Tema proiectului constă în dezvoltarea unei aplicații ce conține o baza de date, creată în sistemul de gestionare a bazelor de date MySQL.

Interfețele vor trebui să permită utilizatorului să execute următoarele operații pe toate tabele: vizualizare, adăugare, modificare și ștergere de date. Vizualizarea tabelelor de legătură va presupune vizualizarea datelor referite din celelalte tabele.

Pentru tema individuală am ales tehnologia FLASK. Asocierea pentru tabela din baza de date este de tipul M:N.

2) Descrierea sistemului de gestiune a bazelor de date MySQL

- Ce este MySQL?

MySQL este un sistem de gestionare a bazelor de date relationale open source care este utilizat în principal pentru aplicațiile online. MySQL poate crea și gestiona baze de date foarte utile (cum ar fi informații despre angajați, inventar și multe altele), la fel ca alte sisteme, cum ar fi popularul Microsoft Access. În timp ce Microsoft Access, MySQL și alte sisteme de gestionare a bazelor de date servesc scopuri similare (de a găzdui datele), utilizarea diferă foarte mult.[1]

MySQL este componenta integrată a platformelor LAMP sau WAMP (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Popularitatea sa ca aplicație web este strâns legată de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. În multe cărți de specialitate este precizat faptul că MySQL este mult mai ușor de învățat și folosit decât multe din aplicațiile de gestiune a bazelor de date, ca exemplu comanda de ieșire fiind una simplă și evidentă: „exit” sau „quit”. [2]

- Cum administrează MySQL bazele de date?

Pentru a administra bazele de date MySQL se poate folosi modul linie de comandă sau, prin descărcare de pe internet, o interfață grafică: MySQL Administrator și MySQL Query Browser. Un alt instrument de management al acestor baze de date este aplicația gratuită, scrisă în PHP, phpMyAdmin.[2]

- De ce este atât de util MySQL?

Baza de date MySQL este folosită în principal ca mijloc de a stoca date pentru aplicații mari, bazate pe web. Site-uri precum WordPress, iStock, GitHub, Facebook, NASA, Marina SUA, Tesla, Scholastic, Spotify, YouTube, Netflix, Glasses Direct, Symantec (și multe altele)

folosesc baza de date MySQL ca mijloc de stocare a datelor pe din interiorul sau exteriorul site-urilor web și serviciilor interne. [1]

3) Tehnologia Flask utilizata in dezvoltarea aplicatiei

- Ce este Flask?

Flask este un microframework web pentru Python, care oferă un mod ușor de a crea aplicații web. Acesta este proiectat pentru a fi ușor de început și de a fi extins în funcție de nevoile proiectului. Flask oferă suport pentru rute, gestionarea sesiunilor și afișarea template-urilor, dar se bazează pe alte biblioteci pentru funcții suplimentare, cum ar fi autentificarea utilizatorilor. Flask este o alegere populară pentru proiecte mici și medii și este ușor de învățat pentru programatorii noi în domeniul web.

4) Limbajul HTML si utilitatea sa in aplicatii

- 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]

- 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 tip front-end, HTML este utilizat impreuna cu CSS și JavaScript.[2]

5) Descrierea Aplicatiei

- Baza de date






Tema individuala se bazeaza pe crearea unei baze de date ce are 2 tabele in asociere M:N. Tabelele sunt: autori si proiecte. Pentru cele 2 tabele am ales câteva attribute caracteristice:

Pentru tabela autori: am ales ca si cheie primara idautori, restul atributelor sunt nume, prenume, adresa, varsta.

Table Name: Schema: **tema_3**

Charset/Collation: Engine: **InnoDB**

Comments:





Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idautori	BIGINT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 nume	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 prenume	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 adresa	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 varsta	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Pentru tabela proiecte: am ales ca si cheie primara idproiecte, restul atributelor sunt: denumire, tematica, termen_limita.

Table Name: Schema: **tema_3**

Charset/Collation: Engine: **InnoDB**

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 idproiecte	BIGINT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 denumire	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 tematica	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 termen_limita	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

- Ce este asocierea M:N?

Asocierea M:N (mai-multi-la-mai-multi) are ca si caracteristica faptul ca fiecarui element inregistrat intr-o tabela ii pot fi asociate mai multe elemente din cealalta tabela si invers. [3]

De exemplu in cazul nostru, un autor poate fi asociat mai multor proiecte, asa cum si un proiect poate fi asociat mai multor autori.

- Ce reprezinta tabela de legatura?

Pentru a crea o relatie mai-multi-la-mai-multi, trebuie sa se creeze o a treia tabela denumita deseori tabela de joctiune, 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 asociere. In aceasta noua tabel, attributele ce au fost selectate ca si chei primare pentru tabelele anterioare vor deveni chei straine (FK) pentru tabela de legatura asociere.



Table Name:

asociere

Schema: **tema_3**

Charset/Collation:

utf8mb4

utf8mb4_0900_ai_ci

Engine:

InnoDB

Comments:

Foreign Key Name	Referenced Table	Column	Referenced Column
fk_asociere_1	`tema_3`.`autori`		
fk_asociere_2	`tema_3`.`proiecte`		

Pentru aceasta tabela am ales „idasociere” ca si cheie primara. Restul atributelor sunt cheile straine idproiecte si idautori.



Table Name:

asociere

Schema: **tema_3**

Charset/Collation:

utf8mb4

utf8mb4_0900_ai_ci

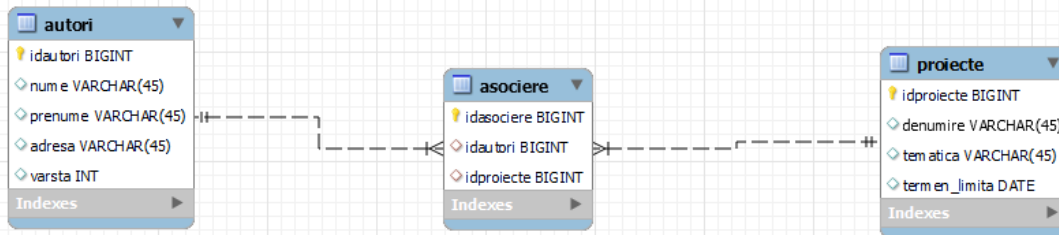
Engine:

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idasociere	BIGINT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
idautori	BIGINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
idproiecte	BIGINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

DIAGRAMA LOGICA A BAZEI DE DATE



De ce este importanta diagrama logica a bazei de date?

Aceasta ofera o reprezentare logica detaliata a datelor celor 3 tabele, a relatiilor dintre ele.

Ce relatii exista intre cele trei tabele?

- Intre autori si asociere exista o asociere de tipul 1:N
- Intre autori si proiecte este de tipul M:N
- Intre proiecte si asociere este de tipul 1:N

FUNCTIONALITATEA APLICATIEI

a) Arhitectura proiectului

Proiectul realizat in tehnologia Flask are urmatoarea structura:

- Un folder templates ce contine paginile html ale aplicatiei.
- Un folder static ce contine poze pentru fundal,css-uri.

b) Implementarea functiilor

Toate paginile html contin structuri de cod html. Acestea sunt accesate de fisiere de tip app.py care se conecteaza independent la baza de date.

De exemplu vom lua tabela proiecte

app2.py se conecteaza prin instructiunile de mai jos:

```
from flask import Flask, render_template, request, url_for, flash
from flask_mysql import MySQL
from werkzeug.utils import redirect
from flask import Blueprint

app = Flask(__name__)
app.secret_key = 'many random bytes'
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '123456'
app.config['MYSQL_DB'] = 'tema_3'

mysql = MySQL(app)
```

Citirea tabelii se face cu instructiunile:

```
@app.route('/')
def Index():
    cur = mysql.connection.cursor()
    cur.execute("SELECT * FROM proiecte")
    data = cur.fetchall()
    cur.close()

    return render_template('proiecte_web.html', proiecte = data)
```


Tabela in pagina web proiecte_web.html:

Python+Flask				
Tabela Proiecte				Adauga Proiect
ID	Denumire	Tematica	Termen limita	
1	Revolutiaa	Istorie	2024-12-01	Editare Eliminare proiect
2	Comunism	Istorie	2023-12-05	Editare Eliminare proiect
3	Colorat	Colorat	2025-08-01	Editare Eliminare proiect
4	Python	Programare	2023-06-23	Editare Eliminare proiect
11	HTML	IT&C	2024-12-02	Editare Eliminare proiect

Adaugarea unui nou proiect se face prin codul de mai jos care trimite o comanda MySQL de tip insert:

```
@app.route('/insert', methods = ['POST'])
def insert():
    if request.method == "POST":
        flash("Proiect postat cu succes!")
        denumire = request.form['denumire']
        tematica = request.form['tematica']
        termen_limita = request.form['termen_limita']
        cur = mysql.connection.cursor()
        cur.execute("INSERT INTO proiecte (denumire, tematica, termen_limita) VALUES (%s, %s, %s)", (denumire, tematica, termen_limita))
        mysql.connection.commit()
        return redirect(url_for('Index'))
```

Stergerea unui proiect se face prin codul de mai jos care trimite o comanda MySQL de tip delete:

```
@app.route('/delete/<string:id_data>', methods = ['GET'])
def delete(id_data):
    flash("Proiect sters!")
    cur = mysql.connection.cursor()
    cur.execute("DELETE FROM proiecte WHERE idproiecte=%s", (id_data,))
    mysql.connection.commit()
    return redirect(url_for('Index'))
```

Modificarea unui proiect se face prin codul de mai jos care trimite o comanda MySQL de tip update:

```
@app.route('/update', methods = ['POST', 'GET'])
def update():
    if request.method == 'POST':
        idproiecte = request.form['idproiecte']
        denumire = request.form['denumire']
        tematica = request.form['tematica']
        termen_limita = request.form['termen_limita']
        cur = mysql.connection.cursor()
        cur.execute("""UPDATE proiecte SET denumire=%s, tematica=%s, termen_limita=%s WHERE idproiecte=%s """, (denumire, tematica, termen_limita, idproiecte))
        mysql.connection.commit()
        flash("Date actualizate.")
        return redirect(url_for('Index'))
```

6) Concluzii

Bazele de date sunt foarte folosite la nivel global de fiecare companie, afacere care se ocupa cu productia are nevoie de baze de date pentru a-si implementa sistemul de lucru. Pentru a face o baza de date cat mai usor de modificat in viitor este important sa avem in vedere realizarea unei arhitecturi ce ofera posibilitatea de a separa functionalitatile.

Baza de date creata si interfata creata cu tehnologia Flask(Python) si prezentata in proiectul de mai sus permite unui utilizator sa efectueze operatiile cerute pe baza de date creata in MySQL: afisare, adaugare, modificare si stergere.

7) Bibliografie

- <https://www.nav.ro/blog/ce-este-mysql/> [1]
- <https://ro.wikipedia.org/> [2]
- Cursuri PIBD [3]
- <https://web.ceiti.md/lesson.php?id=1> [4]
- youtube.com (pentru tot ce tine de programare)