**A circular logo with text and numbers

Description automatically generatedUNIVERSITATEA NAȚIONALĂ DE ȘTIINȚĂ ȘI TEHNOLOGIE POLITEHNICA BUCUREȘTI**

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Proiect (1)**

**Programarea Interfețelor pentru Baze de Date**

Student: Năstase Maria-Magdalena

Grupa: 434D

Profesor coordonator

Ș.l. Dr. Ing. Pupezescu Valentin

București 2025

Cuprins

[1. Cerințe 3](#_Toc188909939)

[2. Tehnologii utilizate 3](#_Toc188909939)

[***2.1*** ***MySQL*** 3](#_Toc188909940)

[***2.2*** ***Pyhton*** 3](#_Toc188909941)

[***2.3*** ***Python Tkinter (tk)*** 4](#_Toc188909942)

[3. Descrierea aplicației 5](#_Toc188909943)

[***3.1*** ***Baza de date*** 5](#_Toc188909944)

***3.2******Diagrama ERD*** 6

[***3.3*** ***Codul*** 7](#_Toc188909945)

*3.3.1* ***Conectarea la baza de date .....................................................................................................................7***

***3.3.2 Gestionarea tabelelor..............................................................................................................................7***

[4. Interfața TK 11](#_Toc188909946)

**5.** **Concluzii**.....................................................................................................................................15

1. **Cerințe**

Să se creeze două aplicații care să conțină o bază de date creată în sistemul de gestiune a bazelor de date MySQL și două interfețe la aceasta (baza de date este comună).

La crearea interfețelor se vor folosi două tehnologii (ex: JSP, Hibernate, JPA, .NET, Python etc.). Baza de date va fi compusă din tabelele: Sculptures și Sculptors, cu asociere de tip M:N.

Interfețele realizate trebuie să permită utilizatorului să execute următoarele operații pe toate tabelele (CRUD): vizualizare, adăugare, modificare și ștergere de date. Vizualizarea tabelelor de legătură va presupune vizualizarea datelor referite din celalte tabele.

Pentru tema primită, se aleg interfețe diferite: tkinter si flask din Python. În prezentul document, se va face referire la tkinter.

## **Tehnologii utilizate**

### ***MySQL***

MySQL este un sistem de gestiune 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 (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 gestiune a bazelor de date servesc scopuri similare (de a găzdui datele), utilizarea diferă foarte mult.

### ***Pyhton***

Python este un limbaj de programare foarte popular, oferind posibilitatea programării structurate, dar și orientate pe obiect, și incluzând și elemente din paradigma funcțională. Este un limbaj de scripting, ceea ce înseamnă că este interpretat și nu compilat, economisind mult timp în procesul de dezvoltare și debugging. Limbajul oferă o multitudine de funcționalități într-o sintaxă ușor de învățat. Are module, clase, excepții, tipuri dinamice și garbage collection. API-ul Python oferă module pentru o gamă foarte mare de funcționalități, de la cele de bază pentru lucrul cu șiruri și fișiere, până la cele pentru lucrul cu procese, threaduri, sockets, serializare etc. Multe dintre aceste module oferă o interfață foarte similară programării la nivel de sistem din C, astfel încât funcțiile Python au același nume si aproximativ aceiași parametrii ca cei pentru funcțiile din C pentru apeluri de sistem.[[1]](#footnote-1)

### ***Python Tkinter (tk)***

Python oferă mai multe opțiuni pentru dezvoltarea GUI (Graphical User Interface). Dintre toate metodele GUI, tkinter este metoda cea mai frecvent utilizată. Este o interfață standard Python pentru setul de instrumente Tk GUI livrat cu Python. Python tkinter este cel mai rapid și mai ușor mod de a crea aplicații GUI.[[2]](#footnote-2)

Ca și în cazul majorității altor legături Tk moderne, Tkinter este implementat ca un wrapper Python în jurul unui interpret Tcl complet încorporat în interpretul Python . Apelurile Tkinter sunt traduse în comenzi Tcl, care sunt transmise acestui interpret încorporat, făcând astfel posibilă amestecarea Python și Tcl într-o singură aplicație.

## **Descrierea aplicației**

### ***Baza de date***

Tema dată presupune asocierea M:N a două tabele, „Sculptures” și „Sculptors”.

Asocierea M:N (Many-to-Many) are ca și caracteristică faptul că fiecărui element înregistrat într-o tabelă îi pot fi asociate mai multe elemente din cealaltă tabelă și invers.

De exemplu, în acest caz, un sculptor poate fi asociat mai multor sculpturi, așa cum și unei sculpturi îi pot fi asociate mai mulți sculptori.

Aplicația are ca scop principal administrarea datelor legate de:

* **Sculptori**: informații precum numele, prenumele, anul nașterii, anul decesului, naționalitatea și stilul artistic.
* **Sculpturi**: titlu, material, înălțime, greutate, anul creației și locația muzeului.
* **Asocieri** între sculptori și sculpturi.

O imagine care conține text, captură de ecran, număr, Font

Descriere generată automat

Figura 1. Tabela Sculptors

O imagine care conține text, software, număr, Font

Descriere generată automat

Figura 2. Tabela Sculptures

Pentru o asociere M:N este necesară crearea unei relații de legătură, în acest caz, „Sculpture\_Sculptor”, o treia tabelă de joncțiune. Scopul acesteia este de a diviza asocierea many-to-many în două asocieri de tip 1:N.

În această nouă tabelă, atributele ce au fost selectate drept chei primare pentru tabelele de mai sus, vor deveni chei străine pentru tabela de legătură.

O imagine care conține text, captură de ecran, Font, linie

Descriere generată automat

Figura 3. Tabela Sculpture\_Sculptor

* 1. ***Diagrama ERD***

O imagine care conține text, captură de ecran, diagramă, Font

Descriere generată automatDiagrama ERD oferă o reprezentare vizuală complexă care ilustrează entitățile, relațiile și constrângerile acestora, oferind o perspectivă detaliată a structurii și interconexiunilor din cadrul bazei de date.

Asocierile dintre tabele:

* Între „Sculptors” și „Sculptures” este o asociere M:N;
* Între „Sculpture\_Sculptor” și „Sculptures” este o asociere 1:N;
* Între „Sculptors” și „Sculpture\_Sculptor” este o asociere 1:N.

### ***Codul***

Conectarea la o bază de date MySQL în Python poate fi realizată folosind diverse module, cum ar fi mysql-connector-python, pymysql sau sqlalchemy. Aceste module facilitează interacțiunea cu baza de date MySQL prin intermediul Python, permițând accesul la date și manipularea acestora.

În paralel, dezvoltarea interfețelor grafice în Python se poate realiza cu ajutorul bibliotecii Tkinter, care oferă un set de instrumente pentru construirea aplicațiilor cu interfață grafică.

* + 1. **Conectarea la baza de date**
* Funcția connect\_to\_database() este utilizată pentru stabilirea conexiunii cu baza de date MySQL.
* Mesajele de eroare sunt afișate utilizatorului în cazul în care conexiunea eșuează.

O imagine care conține text, captură de ecran, Font, software

Descriere generată automat

* + 1. **Gestionarea tabelelor**

**Gestionare sculptori:**

* **Afișare**: show\_sculptors() încarcă toate datele despre sculptori din baza de date și le afișează într-un Treeview.
* **Adăugare**: add\_sculptor() permite utilizatorului să introducă un sculptor nou.
* **Editare**: edit\_sculptor() oferă funcționalitatea de a modifica detalii despre un sculptor existent.
* **Ștergere**: delete\_sculptor() permite eliminarea unui sculptor selectat din bază.

**Gestionare sculpturi:**

* Funcțiile sunt similare celor pentru sculptori: show\_sculptures(), add\_sculpture(), edit\_sculpture() și delete\_sculpture().

**Gestionare asocierile dintre sculptori și sculpturi:**

* **Afișare**: show\_associations() afișează relațiile existente.
* **Adăugare**: add\_association() permite crearea unei noi asocieri între un sculptor și o sculptură.
* **Editare**: edit\_association() modifică o asociere existentă.
* **Ștergere**: delete\_association() șterge relația selectată.

O imagine care conține text, captură de ecran, software

Descriere generată automat

Funcția add\_sculptor() deschide o fereastră pentru adăugarea unui sculptor. Datele introduse sunt colectate din câmpurile de input și inserate în tabela Sculptors din baza de date folosind o interogare INSERT. După adăugare, lista sculptorilor este actualizată și utilizatorului i se afișează un mesaj de succes.

O imagine care conține text, captură de ecran, software

Descriere generată automat

Funcția delete\_sculptor() permite ștergerea unui sculptor selectat din Treeview. ID-ul sculptorului este extras din selecție, iar o interogare DELETE este executată pentru a elimina înregistrarea din tabela Sculptors. După ștergere, lista sculptorilor este actualizată, iar utilizatorului i se afișează un mesaj de succes.

O imagine care conține text, captură de ecran

Descriere generată automat

Funcția edit\_sculptor() permite modificarea detaliilor unui sculptor selectat din Treeview. ID-ul sculptorului este identificat, iar o interogare SQL UPDATE este executată pentru a actualiza valorile din tabela Sculptors pe baza noilor informații introduse de utilizator. După modificare, lista este actualizată, iar utilizatorul primește un mesaj de succes.

**Funcții similare sunt folosite și pentru tabelele Sculptures și Sculpture-Sculptors.**

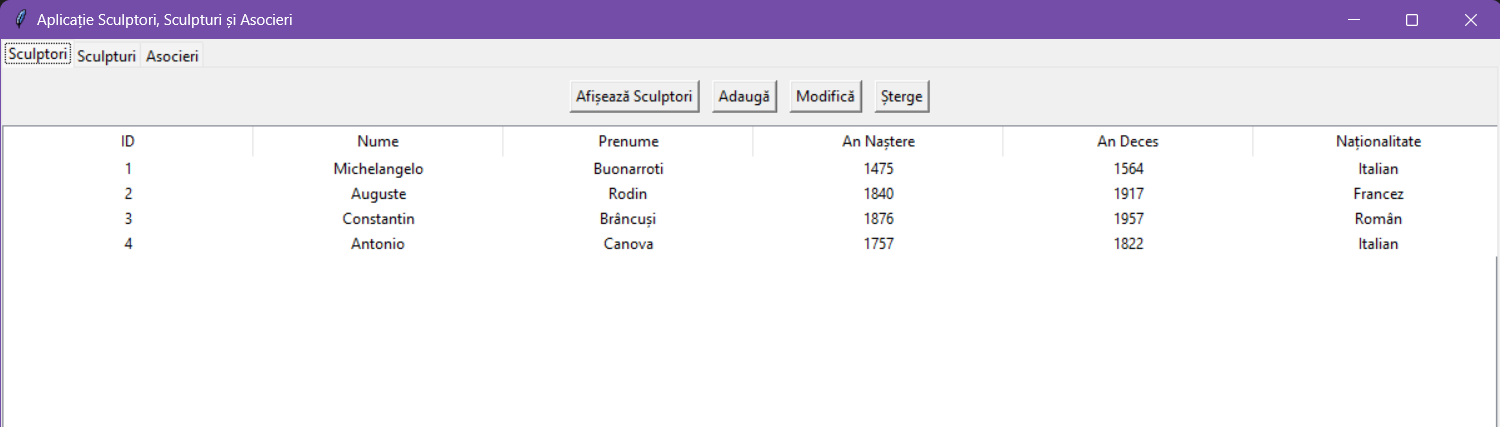
## **Interfața TK**

## **Afișarea datelor**

Fragmentul care permite afișarea și interacțiunea cu datele referitoare la clienți este funcția show\_sculptors(), împreună cu widget-ul Treeview din tab-ul pentru sculptori. Aceasta afișează datele din tabela Sculptors și permite utilizatorilor să selecteze o înregistrare pentru operațiuni suplimentare, cum ar fi ștergerea sau modificarea.

* Afișarea datelor: Funcția show\_sculptors() execută o interogare SQL și populează Treeview-ul cu rezultatele.
* O imagine care conține text, captură de ecran, software, Font

  Descriere generată automatInteracțiunea cu datele: Funcțiile precum delete\_sculptor() și edit\_sculptor() se bazează pe selecția făcută în Treeview[[3]](#footnote-3) pentru a opera asupra datelor specifice.



Formularul pentru afișarea unui tabel

* + 1. **Adăugarea datelor**

O imagine care conține text, captură de ecran

Descriere generată automat

Fragmentul de cod reprezintă logica pentru **adăugarea unui sculptor** în baza de date. Mai exact:

1. **Pregătirea datelor**:
   * Datele introduse de utilizator în câmpurile din interfață (precum entry\_name, entry\_surname etc.) sunt colectate prin metoda .get().
2. **Construirea interogării SQL**:
   * O comandă INSERT INTO Sculptors este pregătită pentru a adăuga datele noi în baza de date, specificând coloanele (Nume, Prenume, etc.) și valorile lor.
3. **Executarea și confirmarea**:
   * Interogarea este executată prin cursorul bazei de date, iar modificările sunt salvate cu connection.commit().
   * Dacă adăugarea are succes, utilizatorului i se afișează un mesaj de confirmare, iar fereastra de adăugare este închisă.
4. **Actualizarea interfeței**:
   * După adăugare, funcția show\_sculptors() este apelată pentru a reîncărca lista sculptorilor afișată în Treeview.

Aceasta este funcția dedicată introducerii de date noi în tabela Sculptors și se asigură că utilizatorul este informat cu privire la succesul sau eșecul operațiunii.

O imagine care conține text, captură de ecran, număr, afișaj

Descriere generată automat

Formularul pentru adăugarea datelor

* + 1. **Modificarea datelor**

O imagine care conține text, captură de ecran, Font, număr

Descriere generată automat

Modificarea datelor se realizează prin selectarea unei înregistrări și apăsarea butonului de **Modificare** din interfață. Fragmentul de cod care preia datele din câmpurile introduse de utilizator și le utilizează pentru actualizarea sculptorului selectat în baza de date este ilustrat mai sus.

* + 1. **Ștergerea datelor**

Ștergerea datelor se realizează prin selectarea liniei și apăsarea butonului de Șterge din pagină.

O imagine care conține text, software, Pictogramă computer, Software multimedia

Descriere generată automat

**5. Concluzii**

Proiectul dezvoltat utilizează tehnologiile MySQL și Python Tkinter pentru a crea o aplicație robustă și ușor de utilizat pentru gestionarea datelor despre sculptori și sculpturi. Cu ajutorul acestei aplicații, utilizatorii pot administra eficient baza de date, având posibilitatea de a efectua operațiuni de creare, citire, actualizare și ștergere a informațiilor referitoare la sculptori, sculpturi și asocieri între acestea.

Interfața grafică dezvoltată cu Python Tkinter oferă utilizatorilor o experiență intuitivă și accesibilă. Designul simplu și funcțional permite navigarea facilă și interacțiunea directă cu datele stocate în baza de date MySQL.

Prin intermediul operațiunilor CRUD (creare, citire, actualizare, ștergere), utilizatorii pot gestiona cu ușurință informațiile din baza de date, având posibilitatea de a adăuga noi înregistrări, de a vizualiza și edita datele existente, precum și de a șterge informațiile care nu mai sunt necesare.

Unul dintre avantajele majore ale acestei aplicații constă în utilitatea sa pentru organizarea și administrarea informațiilor legate de sculpturi și sculptori. Aplicația oferă o platformă centralizată pentru gestionarea acestor date, facilitând procesul de evidență și monitorizare, contribuind astfel la eficientizarea activităților administrative.

În concluzie, prin combinarea tehnologiilor MySQL și Python Tkinter, proiectul oferă o soluție completă și simplă pentru gestionarea informațiilor despre sculptori și sculpturi. Interfața intuitivă și funcționalitatea extinsă fac din această aplicație un instrument esențial pentru utilizatori în activitățile lor zilnice.

**Bibliografie**

[**https://docs.python.org/3/library/tk.html**](https://docs.python.org/3/library/tk.html)

[**https://docs.python.org/3/library/tkinter.html**](https://docs.python.org/3/library/tkinter.html)

[**https://www.geeksforgeeks.org/python-gui-tkinter/**](https://www.geeksforgeeks.org/python-gui-tkinter/)

[**https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/index.html**](https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/index.html)

1. https://ocw.cs.pub.ro/courses/asc/laboratoare/01 [↑](#footnote-ref-1)
2. https://www.geeksforgeeks.org/python-gui-tkinter/ [↑](#footnote-ref-2)
3. Un **Treeview** este un widget din biblioteca **Tkinter** folosit pentru afișarea organizată a datelor sub formă de tabel sau structură ierarhică (similar unui arbore). Este foarte util pentru afișarea listelor, tabelelor sau datelor complexe care pot avea relații ierarhice. [↑](#footnote-ref-3)