



UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI

DISCIPLINA: BAZE DE DATE

Evidența pacienților unui cabinet de psihiatrie

Coordonator,

Prof. Cristian-Nicolae Buțincu

Student,

Boacă Mădălina-Elena

Grupa 1307A

Iași, 2022

Descrierea proiectului

Aplicația are ca scop modelarea datelor unui cabinet de psihiatrie, acestea constând din fise medicale, datele pacienților, medici ce lucrează la acest centru, medicamentele prescrise sau afecțiunile întâlnite.

Centrul îl constituie **fisele medicale** care conțin informații despre pacient, data eliberării, data la care se va efectua următorul control, medicamente prescrise, afecțiunile de care suferă pacientul și medicul psihiatru din centru care a eliberat fisa. Pentru a facilita căutarea fiselor ele vor fi indexate. **Pacienți** sunt definiți prin nume, prenume, data nașterii, număr de telefon, gen, adresa. **Medici psihiatrii** sunt identificați pe baza unui ID unic, nume, prenume, număr de telefon și o eventuală adresă de email. **Medicamentele** prezintă, de asemenea, un ID unic, nume și o firmă producătoare presupunând că un medicament poate fi produs în același timp de mai multe companii, iar medicul preferă o companie pe anume pe criterii de experiență sau calitate. **Afecțiunile** vor fi de asemenea indexate pentru a putea fi referite mai rapid în cazul în care denumirea este riguroasă. Pentru a se putea obține o imagine mai clară a afecțiunilor pacientului se vor furniza detalii ce pot conta pentru suferind sau familia lui în încercarea de tratare a bolii, de diminuare a simptomelor sau evitarea unor situații nefericite. Acestea sunt gravitate, rata de vindecare, necesitatea unui tratament medicamentos (nu toate afecțiunile implică și un astfel de element), rata suicid (acest gest necugetat fiind des întâlnit în astfel de cazuri apare necesitatea de a avea o grijă mai mare de astfel de pacienți), specificitate gen, asistența personală.

Apariția unui nou pacient are ca efect începerea completării unei noi fise medicale și consultul lui de către un medic. Medicul va evalua starea pacientului, are sau nu are o problemă, are o singură afecțiune sau o combinație. În funcție de aceste date va stabili dacă este sau nu necesar un tratament medicamentos și în caz afirmativ va prescrie unul indicând numele medicamentului, o firmă producătoare recomandată, un număr de comprimate zilnice și un orar/mod de administrare. Tot în urma consultului va decide și următoarea dată a întâlnirii. Pentru eventuale întrebări pe perioada tratamentului atât medicul va putea contacta pacientul cât și pacientul medicul pentru eventuale neclarități, existând date de contact pentru acest fapt.

Evidența unor afecțiuni în informațiile cabinetului are ca scop posibilitatea verificării experienței în tratarea anumitor patologii, putând contoriza numărul de cazuri tratate/aflăte în tratament pentru fiecare tulburare. Totodată se dorește și evidența medicamentelor prescrise pentru a putea vizualiza cele mai dese produse recomandate și firme, medicul putând pe baza unei analize să prescrie remedii mai bune.

Solicitarea unor detalii despre afecțiune este fundamentală pentru acțiunile pacientului, apropiatilor lui în încercarea de tratare/diminuare a tulburărilor precum și a deciziilor legate de supraveghere permanentă.

Necesitatea **modificării datelor** apare pentru pacienți atunci când datele lor personale se schimbă (schimbare nume de familie, adresa, număr de telefon etc.) și sunt absolut necesare mai ales când se menține o colaborare permanentă cu psihiatrul. Totodată același lucru se întâmplă și

în cazul medicului. Medicamentele își pot schimba de asemenea denumirea. Detaliile afecțiunilor se schimbă des mai ales în cazul unor tulburări noi pe parcursul unor descoperiri. Actualizarea unei fișe medicale este fundamentală mai ales în cazul în care se constată o greșeală imediat după completare.

Ștergerea unor pacienți ca urmare a încetării colaborării, a unor medicamente la dispariția lor de pe piață, a unor psihiatri la încetarea activității sau a unor fișe vechi din dorința de a nu păstra date inutile sunt de asemenea operații fundamentale.

Însă, poate, cel mai mult contează capacitatea de a viziona datele în diverse forme după necesități. Din multitudinea de posibilități amintesc: vizualizarea tuturor pacienților, medicilor psihiatri, afecțiunilor tratate, detaliilor tulburărilor, medicamentelor, fișelor medicale; medicamentelor luate de un anumit pacient în acest moment, a afecțiunilor de care suferă acum; pacienților tratați în acest moment de un anumit medic, medicamentelor ce sunt administrate în acest moment, istoricului afecțiunilor unui pacient, a tratamentelor pe care le-a luat etc.

Tehnologii folosite

A. FRONT-END

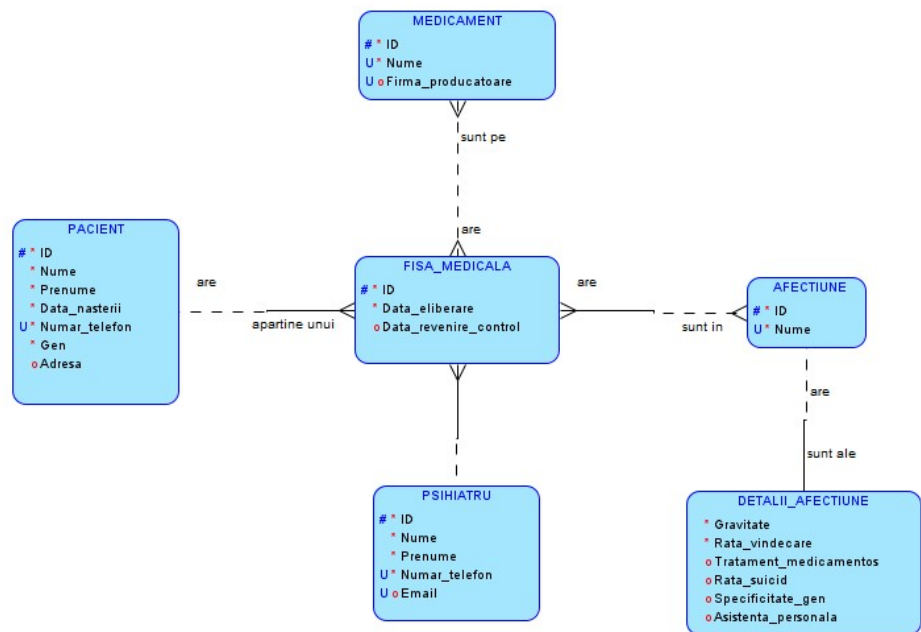
Partea de interfață pentru comunicarea cu baza de date a fost realizată în limbajul **Python** utilizând pachetele: `cx_oracle` (pentru conectarea la baza de date de pe server-ul facultății), `PIL` și `tkinter` pentru grafică.

B. BACK-END

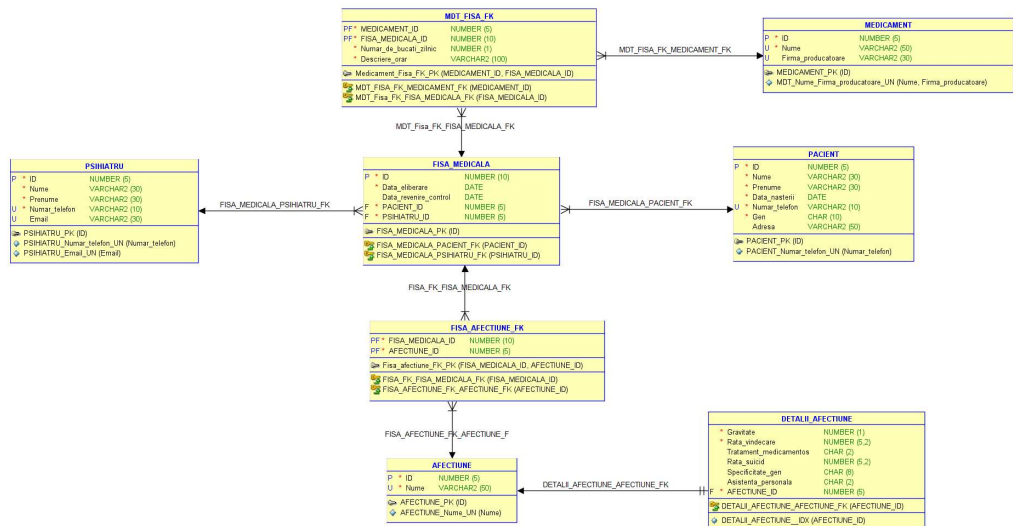
S-a utilizat limbajul SQL împreună cu Oracle SQL Developer (scripturi de inserare, creare baze de date din codul generat), Oracle SQL Developer Data Modeler (creare model logic și relational).

Structura și inter-relaționarea tabelor

Modelul logic



Modelul relațional



Aplicația va porni în modelul logic de la 6 entități:

- FISA_MEDICALĂ
- PACIENT

- PSIHIATRU
- AFECTIUNE
- MEDICAMENT
- DETALII_AFECTIUNE

Între entități există 5 relații (o relație one to one, două relații many to one și două relații many to many - pentru normalizarea datelor) din necesități ca un pacient să poată avea mai multe fișe medicale (many to one), un psihiatru să poată scrie mai multe fișe medicale (many to one), un medicament să poată fi prescris mai multor pacienți și un pacient să primească mai multe medicamente (many to many), sau o afecțiune să fie comună mai multor pacienți și un pacient să poată suferi de mai multe tulburări (many to many).

Legăturile se realizează astfel: între PACIENT și FISA MEDICALĂ pe baza ID din pacient și a PACIENT_ID din fișa medicală, între fișa medicală și psihiatru pe baza ID din psihiatru și a PSIHIATRU_ID din fișa medicală. Pentru crearea relației many to many dintre o fișă medicală și un medicament se va insera în modelul relațional tabel MDT_FISA_FK unde MDT menționează denumirea scurtă a cuvântului medicament, legătura fiind făcută în noua tabelă pe baza unui index legat la indexul din tabela medicament și a unui alt index legat la tabela fișă medicală (se pot observa în imaginea cu modelul relațional). O abordare similară este urmată și pentru scrierea mai multor afecțiuni pe aceeași fișă, în acest caz inserându-se tabela FISA_AFECTIUNE_FK cu doi indexi ce referă un element din fișa medicală respectiv afecțiune. Relația dintre afecțiune și detalii se face pe baza coloanei AFECTIUNE_ID adăugată în tabela detalii afecțiune.

Mai este de remarcat că în tabela MDT_FISA_FK s-au adăugat doi membri pentru numărul de comprimate zilnic și modul de administrare pentru a specifica numărul de comprimate zilnice pe care pacientul trebuie să le ia și modul de administrare.

Descrierea constrângerilor

Constrângerile la nivelul tabelii **Pacient** sunt: ID-ul trebuie să fie primary key, deoarece este unic pentru fiecare pacient și este folosit la legarea pacientului de fișele medicale, numele și prenumele nu trebuie să fie NULL, să nu aibă o dimensiune vidă și să nu conțină decât litere. Data nașterii va trebui să fie mai mică decât data curentă și să nu fie NULL-ă. Numărul de telefon nu poate conține cifre și trebuie să fie unic, pentru a identifica mai ușor un pacient (numele, prenumele, data nașterii se pot repeta). Gen-ul nu poate fi NULL și trebuie ales din „Feminin” sau „Masculin”. Pentru adresă nu s-au aplicat constrângeri. Pentru verificarea datei de naștere s-a folosit trigger.

Fișă medicală are ca și constrângeri ID de tip primary key pentru legătura cu celelalte tabele, data următorului control poate fi NULL dacă pacientul s-a vindecat, data eliberării este obligatorie.

Tabela **Psihiatru** are ID- cheie primara după aceeași logica enunțata mai sus, numele, prenumele, numărul de telefon sunt NOT NULL. Numele si prenumele nu pot conține 0 caractere, iar numărul de telefon nu poate conține litere. Numărul de telefon si adresa de email trebuie sa fie de asemenea unice.

Tabela **Medicamente** are doua constrângeri de tip check pentru verificarea lungimi numelui si a firmei(sa fie mai mare decât 1) și un primary key pentru ID. Pentru identificare unica s-a definit o constrângere unique pe doua coloane(nume, firma producătoare).

În tabela **Afecțiune** întâlnim arhicunoscutul ID având constrângerea primary key și nume ce nu poate fi NULL si nici avea o dimensiune mai mica decât 1. Numele unei afecțiuni este unic (unique).

Detalii afecțiune conține mai multe constrângeri spre deosebire de celelalte tabele. Gravitate este un număr din gama 0-9, rata de vindecare este un număr încadrat între 0.00 si 100.00, coloana tratament medicamentos poate lua doar valorile „Da” și „Nu”. Rata de suicid se încadrează tot între 0.00 si 100.00. Coloana specificitate gen poate lua una din valorile : „Feminin” , „Masculin ” sau „Nu”. Coloana asistenta personala poate fi „Da” (pacientul are nevoie de îngrijire personala) sau „Nu” (pacientul se poate descurca singur).

Conectarea la baza de date din aplicație

Conectarea la baza de date s-a făcut utilizând pachetul(modulul) cx-oracle și Oracle client. Conectarea a necesitat includerea unui folder cu diferite fișiere descărcat de la companie. Au fost furnizate tot o data și informații cu privire la server-ul de conectare și la datele de autentificare.

```
user = 'bd034'
password = 'bd034'
cx_Oracle.init_oracle_client(
    lib_dir=r"C:\Users\boaca\Desktop\Facultate\BDP\Interfata\InterfataBD\instantclient_21_7")
dsn_tns = cx_Oracle.makedsn('bd-dc.cs.tuiasi.ro', '1539', service_name='orcl')
self.conn = cx_Oracle.connect(user=user, password=password, dsn=dsn_tns)
self.cursor = self.conn.cursor()
```

Interfața aplicației și comenzi SQL

Operațiile au fost realizate în format grafic pentru toate cele șase tabele, însă aici se va prezenta doar un exemplu pentru fiecare operație, cel pe care l-am considerat cel mai semnificativ.

Adăugare unei fise medicale

Pentru adăugarea unui noi fise pacientul și psihiatrul au fost aleși dintr-un meniu popup și au fost identifica-ti prin nume, prenume și numărul de telefon(numărul de telefon este unic). S-au putut alege mai multe medicamente precum și afecțiuni. În procesul de adăugare al unei noi fise s-au modificat 3 tabele FISA_MEDICALA, MDT_FISA_FK, FISA_AFECTIUNE_FK. Pentru verificarea corectitudinii datelor introdu-se s-au tratat excepțiile generate de executarea comenzilor SQL. În cazul validității/invalidității datelor introdu-se afișându-se ferestre popup în concordanta.

Acasa

Fise pacienti

Pacienti

Medicamente

Afectiuni

Psihiatri

Detalii afectiuni

Fise medicale

Adauga

Adaugare fisa noua

Pacient

Lungu Iile 0751435678

Medic psihiatru

Munteanu Marius Calin 033341234

Data eliberare fisa

6.12.2022

Data revenire control

6.6.2023

Medicamente

Enxiron de la Antibiotice lasi

Epitol de la Sanofi

Ergonil de la Sanofi

Delapone de la JSCA

Delapone de la Sanofi

Afectiuni

Delir

Depresie

Acrofobie

ADHD

Autism

Se vor introduce in ordine pentru fiecare medicament pe o linie cantitatea si programul.

Cantitate

1

2

Mod administrare


1 dimineata

1 dimineata si una seara

Notă

Datele au fost inserate cu succes!

OK



Acasa

Fise pacienti

Pacienti

Medicamente

Afectiuni

Psihiatri

Detalii afectiuni

Fise medicale

Adauga

Adaugare fisa noua

Pacient

Lungu Iile 0751435678

Medic psihiatru

Aurmel Andreia Elena 0703341234

Data eliberare fisa

12.12.2022

Data revenire control

12.12.2021

Medicamente

Enxiron de la Antibiotice lasi

Epitol de la Sanofi

Ergonil de la Sanofi

Delapone de la JSCA

Delapone de la Sanofi

Afectiuni

Delir

Depresie

Acrofobie

ADHD

Autism

Se vor introduce in ordine pentru fiecare medicament pe o linie cantitatea si programul.

Cantitate

1


Mod administrare

seara laolalta de culcare

Eroare

Datele introduse nu sunt valide!

OK



Exemple de comenzi SQL folosite(la inserarea unei fise):

```
app.cursor.execute(
    'insert into fisa_medicala values(null,\'' + app.sheet.intrareDataEliberare + '\',\'\' + app.sheet.intrareControlUrmator + '\',\'\' +
    str(idpacient[0][0]) + ',\'\' + str(app.sheet.intrareIDMedic) + '\')')
app.cursor.execute('commit')
```

Valoarea null, este pentru ID-ul fisei, întrucât el a fost generat în tabele cu auto increment.

```
app.cursor.execute('insert into MDT_FISA_FK values(' + str(listMed[i]) + ',\'\' + str(
    ind[0][0]) + ',\'\' + str(int(cantitati[i])) + '\',\'\' + programe[i] + '\')')
```

```
app.cursor.execute('insert into FISA_AFECTIONE_FK values(' + str(ind[0][0]) + ',\'\' + str(i) + '\')')
```

Valorile inserare au fost extrase din contoarele din interfață :

```
app.sheet.intrarePacient = app.sheet.strPacient.get()
app.sheet.intrareMedic = app.sheet.strMedic.get()
app.sheet.intrareDataEliberare = app.sheet.dataEliberareT.get("1.0", "end")
app.sheet.intrareControlUrmator = app.sheet.revenireControlT.get("1.0", "end")
app.sheet.intrareMedicamente = []
cantitate = app.sheet.cantitateT.get("1.0", "end")
for i in app.sheet.medicamenteLB.curselection():
    app.sheet.intrareMedicamente.append(i)
app.sheet.intrareAfectiuni = []
for i in app.sheet.afectiuniLB.curselection():
    app.sheet.intrareAfectiuni.append(i)
app.sheet.intrareOrar = app.sheet.detaliiT.get("1.0", "end")
```

Vizualizarea pacienților

[Acasa](#)[Fise pacienti](#)[Pacienti](#)[Medicamente](#)[Afectiuni](#)[Psihatri](#)[Detalii afectiuni](#)

Pacienti

Visualizare

[Visualizare pacienti](#)

ID	Nume	Prenume	Data nasterii	Numar de telefon	Gen	Adresa
100	Toma	Gheorghe	1990-04-03 00:00:00	0756435678	Masculin	Nr 12, Strada Panselutelor, Iasi
110	Lungu	Ilie	1995-01-01 00:00:00	0751435573	Masculin	Nr 1, Strada Revolutiei, Iasi
120	Ilie	Victor	1981-12-10 00:00:00	0751435222	Masculin	Nr 13, Strada Revolutiei, Iasi
130	Mihnea	Maria	1970-04-11 00:00:00	0751455578	Feminin	Nr 102, Strada Florilor, Pascani
140	Orban	Irina	1977-05-20 00:00:00	0754455578	Feminin	Nr 342, Strada Gloriei, Costesti
150	Robu	Ion	1977-06-25 00:00:00	0754466573	Masculin	Nr 42, Strada Golia, Iasi
160	Robu	Ion	1979-06-15 00:00:00	0756666578	Masculin	Nr 2, Strada Ion Neculcea, Letcani
170	Robu	Mihai	1960-09-25 00:00:00	0754674534	Masculin	Nr 56, Strada Golia, Iasi
180	Airinei	Elena	2000-11-05 00:00:00	0712374534	Feminin	Bloc C1, Nr 31, Strada Tudor Vladimirescu, Iasi
190	Bran	Mirela	1990-10-12 00:00:00	0787904534	Feminin	Bloc B2, Nr 12, Strada Victoriei, Targu Frumos
200	Bran	Mirela	1997-10-13 00:00:00	0767994534	Feminin	Bloc B2, Nr 12, Strada Victoriei, Targu Frumos

[Pacienti anteriori](#)[Urmatori pacienti](#)

Cele doua butoane au sens atunci când numărul pacienților crește.

```
try:
    app.cursor.execute('select * from pacient order by ID')
    rows = app.cursor.fetchall()
    name = ('ID', 'Nume', 'Prenume', 'Data nasterii', 'Numar de telefon', 'Gen', 'Adresa')
    # print(rows)

    nrelemente = int(app.height / (12 * 3))

    if app.patient.page > (len(rows) / nrelemente):
        app.patient.page = int(len(rows) / nrelemente)

    if app.patient.page < 0:
        app.patient.page = 0

    indexStat = app.patient.page * nrelemente
    indexStop = app.patient.page * nrelemente + nrelemente

    if indexStop > len(rows):
        indexStop = len(rows)

    for j in range(7):
        e = tk.Entry(app, width=(int(app.width / 50)), fg='blue',
                     font=('Arial', 12, 'bold'), background='white')
        e.place(x=j * (app.width / 10) + (app.width / 10), y=((app.height / 3) - (int(app.width / 100))))
        e.insert(tk.END, name[j])
        app.patient.table.append(e)

    for i in range(indexStat, indexStop):
        for j in range(7):
            if j == 6:
                e = tk.Entry(app, width=(int(app.width / 50)), fg='LightBlue',
                             font=('Arial', 12, 'bold'), background='gray10')
            else:
                e = tk.Entry(app, width=(int(app.width / 90)), fg='LightBlue',
                             font=('Arial', 12, 'bold'), background='gray10')
            e.place(x=j * (app.width / 10) + (app.width / 10),
                  y=(i % nrelemente) * (int(app.width / 100)) + (app.height / 3))
            try:
                e.insert(tk.END, rows[i][j])
                app.patient.table.append(e)
            except:
                app.patient.table.append(e)
```

Modificarea psihiatrilor

[Acasa](#) [Fise pacienti](#) [Pacienti](#) [Medicamente](#) [Afectiuni](#) [Psihiatri](#) [Detalii afectiuni](#)

Medici psihiatrii

Modificare

[Modificare date psihiatru](#)

Nume*

Ionciu

Prenume*

Ioana


Telefon*

0233467671

Email

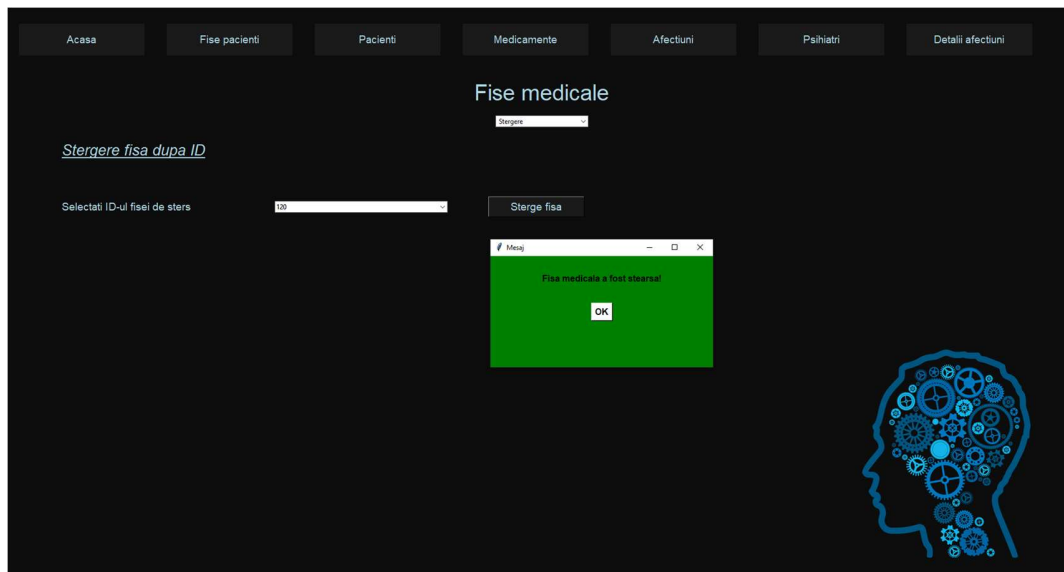
ioanaTurcu0@gmail.com

OK



```
app.cursor.execute('update psihiatru set nume = \'' + app.doctor.insertNume + '\' where numar_telefon=' + str(id))
app.cursor.execute(
    'update psihiatru set prenume = \'' + app.doctor.insertPrenume + '\' where numar_telefon=' + str(id))
app.cursor.execute(
    'update psihiatru set email = \'' + app.doctor.insertEmail + '\' where numar_telefon=' + str(
        id))
app.cursor.execute(
    'update psihiatru set numar_telefon = \'' + app.doctor.insertTelefon + '\' where numar_telefon=' + str(
        id))
```

Ștergerea unei fișe este de asemenea relevantă, având în vedere că presupune modificarea a trei tabele.



```
def stergereFisa():
    nume = app.sheet.stergereC.get()
    app.patient.pacientC.set('')
    try:
        app.cursor.execute('delete from fisa_medicala where id = ' + nume)
        app.cursor.execute('commit')
        open_popup("Datele au fost sterse!")
    except IndexError as e:
        open_popupError('Selectati un element de sters!\nCasetă vida!')
        print(e)
    except cx_Oracle.DatabaseError as e:
        print(e)
        if str(e).__contains__('ORA-00936: missing expression'):
            open_popupError('Selectati un element de sters!\nCasetă vida!')
        if str(e).__contains__('FISA_AFECTIUNE_FK_FISA_FK') or str(e).__contains__('FISA_FK_FISA_MEDICALA_FK'):
            app.cursor.execute('delete from MOT_FISA_FK where FISA_MEDICALA_ID = ' + nume)
            app.cursor.execute('commit')
            app.cursor.execute('delete from FISA_AFECTIUNE_FK where FISA_MEDICALA_ID = ' + nume)
            app.cursor.execute('commit')
            app.cursor.execute('delete from fisa_medicala where id = ' + nume)
            app.cursor.execute('commit')
            open_popup('Fisa medicala a fost stearsa!')
            selectOperatieFise(3)
    except Exception as e:
        print(e)
    app.sheet.valori = []
    app.cursor.execute('select id from fisa_medicala order by id asc')
    rows = app.cursor.fetchall()
    app.sheet.valori = []
    for i in rows:
        app.sheet.valori.append(i[0])

    app.patient.pacientC['values'] = app.sheet.valori # se vor citi din baza de date
    selectOperatieFise(3)
```

Pentru tratarea constrângerilor în interfața grafică au fost gestionate excepțiile generate (constrângeri de valori imposibile, câmpuri invalide, valori neunice pentru câmpuri, chei externe etc).

Exemplu pentru adăugarea unui pacient:

```
except cx_Oracle.IntegrityError as e:
    if str(e).__contains__('PACIENT_NUME_CK'.upper()):
        open_popupError("Numele este gresit!")
    elif str(e).__contains__('PACIENT_Prenume_CK'.upper()):
        open_popupError("Prenumele este gresit!")
    elif str(e).__contains__('pacient_gen'.upper()):
        open_popupError("Genul este gresit!")
    elif str(e).__contains__('pacient_numar_telefon_ck'.upper()):
        open_popupError("Numarul de telefon este invalid!")
    elif str(e).upper().__contains__('ORA-01400'.upper()):
        open_popupError("Exista campuri vide!")
    elif str(e).upper().__contains__('ORA-01843: not a valid month'.upper()):
        open_popupError("Luna nu este valida!")
    print(e)
    traceback.print_exc()
except cx_Oracle.DatabaseError as e:
    if str(e).__contains__('TRG_PACIENT_BRIU'.upper()):
        open_popupError("Data de nastere invalida.\n Trebuie sa fie mai mica decat data curenta!")
    elif str(e).upper().__contains__('ORA-01830'.upper()) or str(e).upper().__contains__('ORA-01858'.upper()):
        open_popupError("Format data necorespunzator.\nFolositi DD-MM-YYYY!")
    elif str(e).upper().__contains__('ORA-01843: not a valid month'.upper()):
        open_popupError("Luna nu este valida!")
    elif str(e).upper().__contains__(
        'ORA-12899: value too large for column "BD034"."PACIENT"."NUMAR_TELEFON".upper()):
        open_popupError("Numarul de telefon nu este valid!")
    traceback.print_exc()
```

Descrierea tranzacției

La fiecare modificare a bazei de date se va efectua „commit” pentru a fi vizibile modificările pentru alte operații.