



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

DISCIPLINA BAZE DE DATE

Site de vanzari pentru masini si unelte de atelier

Coordonator,
Mironeanu Catalin

Student,
Chirica Andrei – 1309A

❖ **Proiectarea aplicatie Back-End:**

Aplicatia are rolul de a face data entry pentru un site ce vinde utilaje si unelte pentru atelier. In baza de date sunt stocate proprietati ale fiecarui produs, descrieri, producatorii si tara lor de origine si categoriile, respectiv subcategoriile din care fac parte produsele.

Fiecare categorie(ex: utilaje pentru prelucrat lemn sau metal) are fie o subcategorie(ex: categoria insasi) fie mai multe subcategorii(ex: masini frezat, exhaustoare pentru categoria lemn).

Fiecare subcategorie are unul sau mai multe produse.

Producatorii sunt cei care creeaza respectivele produse. Ei pot avea o descriere sau nu si poate fi precizata tara de provenienta.

Detaliile au o descriere amanuntita a produsului si o poza stocata sub forma unui site.

Produsele in sine au un set de proprietati ce vor ajuta clientii sa se decida in alegerea celui mai bun din ele.

❖ **Structura proiectului:**

In crearea bazei de date s-au folosit urmatoarele entitati:

- Categorie: contine denumirea categoriei si o mini descriere
- Subcategorie: contine denumirea subcategoriei si o mini descriere
- Producator: stocheaza denumirea producatorului, tara de origine si o descriere.
- Detalii: au o descriere amanuntita a produsului cu cateva caracteristici ce nu se pot incadra in proprietati plus o poza a produsului stocata sub forma un site.
- Produsul: au denumirea, pretul si stocul produsului.
- Proprietati: contin reuniunea tuturor denumirilor proprietatilor

- Val_prop: stocheaza orice valoare a unei proprietati ce poate exista in baza de date.
- Prod_prop: in sine contine doar id-ul cheie primara, insa de fapt are rol de tabela de legatura produse-proprietati-valori. Contine o cheie straina a produsului, o cheie straina a proprietatii si o cheie straina a valorii proprietatii. In acest fel se poate construi orice fel de combinatie necesara.

❖ **Relatiile dintre entitati:**

In principiu sunt prezente doar legaturi de 1:1 si 1:m, insa era prezenta si o relatie de m:m care din motive de design a fost redusa la mai multe relatii de 1:m.

- Categorie si Subcategorie(1:m): Avem mai multe categorii, insa fiecare categorie trebuie impartita in subcategorii pentru a facilita procesul de cautare a clientului.
- Subcategoria si Produs(1:m): Fiecare subcategorie are o gama de produse care in esenta fac acelasi lucru insa au manufacturi diferite, materiale de constructie diferite, producatori diferiti etc.
- Producatori si Produs(1:m): Un singur producator poate produce o multime de produse din categorii, respectiv subcategorii diferite.
- Detalii si Produse(1:1): Fiecare produs are o descriere si poza diferita. Si fiecare descriere si poza este diferita pentru fiecare produs in parte.
- Produse, prod_prop, val_prop si Proprietati(1:m / m:m): In mod normal Produse si Proprietati ar trebui sa fie in relatie de m:m in modelul logic, insa atunci ar face imposibila legarea entitatii val_prop. Din acest motiv a fost creata entitatea prod_prop ca tabela de intermediara pentru celelalte 3 entitati.

❖ **Constrangeri, primary keys, foreign keys, unique keys autoincrements, checks:**

→ Primary keys:

- ID_Cat(Categorie)
- ID_Sub(Subcategorie)
- ID_Prod(Produs)
- ID_Producator(Producator)
- ID_Desc(Detalii)
- ID_Prop(Proprietati)
- ID_Prod_Prop(Prod_Prop)
- ID_val_prop(val_prop)

→ Foreign keys:

- ID_Cat(Subcategorie)
- ID_Sub, ID_Producator, ID_Desc(Produs)
- ID_Prod, ID_prop, ID_val_prop(Prod_Prop)

→ Unique keys:

- Denumire_Cat(Categorie): Nu pot exista doua categorii cu aceeasi denumire.
- Denumire_Sub(Subcategorie): Aceeasi regula ca la categorii
- Denumire_Producator(Producator): Nu pot exista doi producatori cu aceeasi denumire.
- Denumire_Prod(Produs): Denumirea produselor vor diferi intotdeauna intrucat ele au in denumire si un cod de model.
- Denumire_Prop(Proprietati): Toate proprietatile din baza de date sunt stocate aici impreuna, neexistenta constrangerii unique ar ingreuna procesul de data entry.
- Valoare(Val_prop): Aceeasi situatie ca la proprietati.

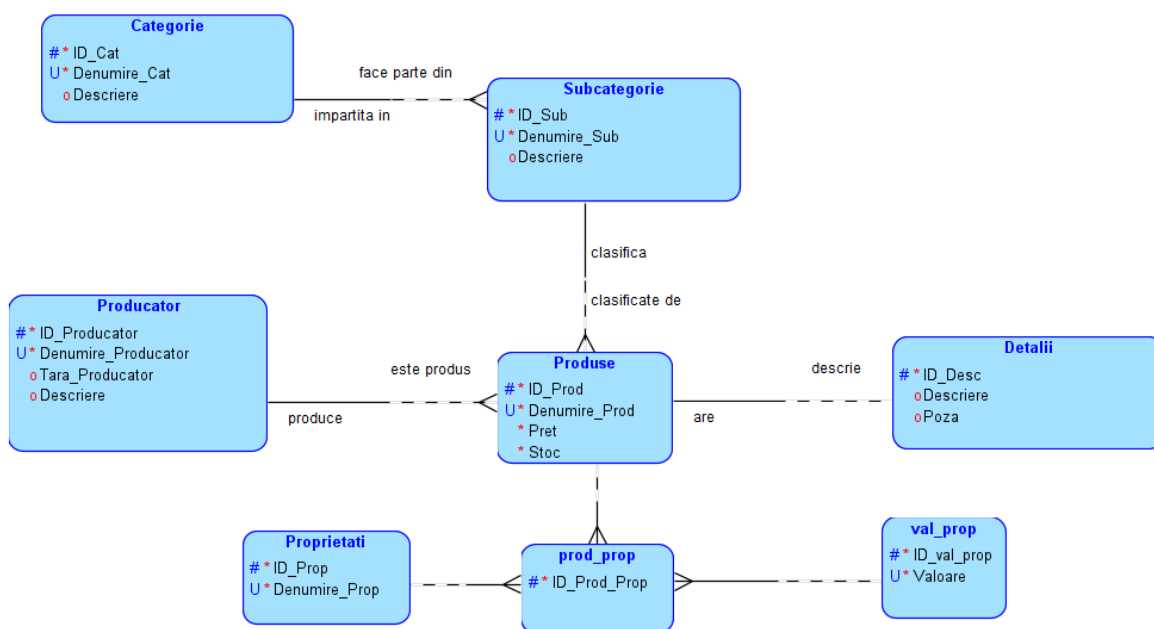
→ Constrangeri/ checks:

- Denumire are o constrangere de lungime, sa nu fie null si un check de tip regex pentru a verifica faptul ca denumirea nu contine cifre. Valabil pentru toate denumirile. Exceptie facand Denumire_Prod si

Denumire_Prop. Ele au constrangerea de tip regex ca sa poate include caractere alfanumerice, unele caractere speciale si spatiu.

- Valoare: constrangere de lungime si verificare de tip regex pentru continut de caractere alfanumerice, unele caractere speciale si spatiu.
- Poza contine check pentru a verifica daca forma stringului arata ca un site de tip „http(s)://”.

❖ Modelul logic:

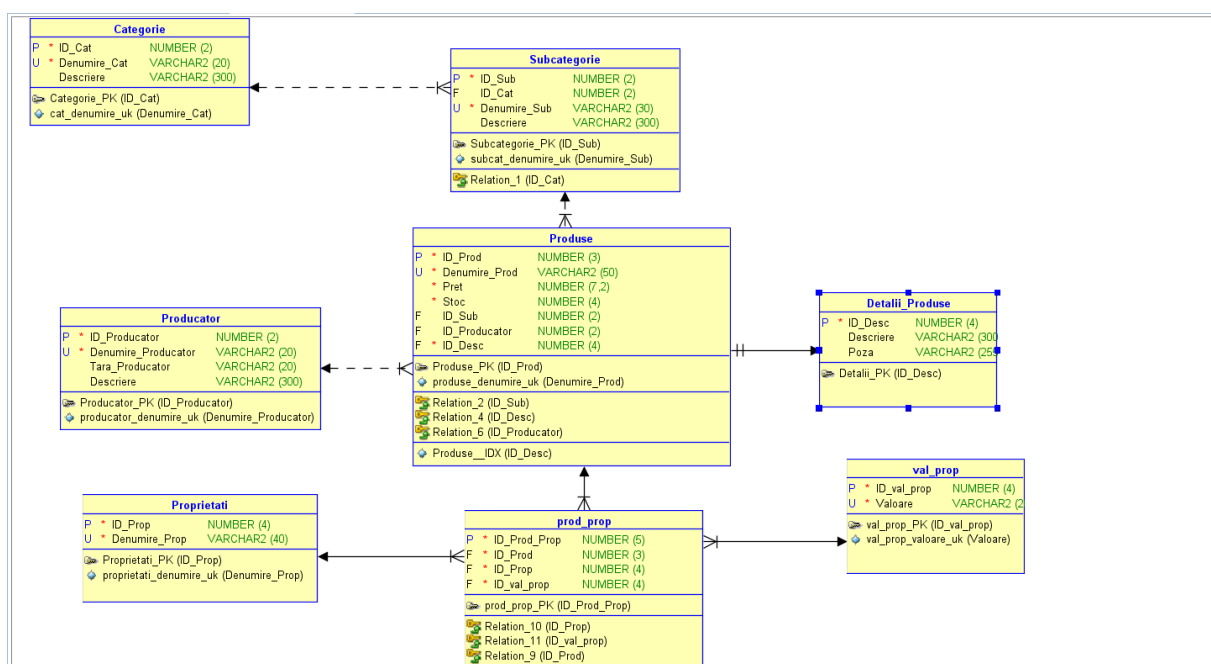


❖ Modelul relational:

Normalizarea tabelelor:

→ Toate tabelele sunt deja in a treia forma normala deoarece:

- Un atribut contine valori atomice din domeniul sau si nu contine grupuri care se repeta(ex. in tabela produse pentru fiecare cheie id_prod exista o singura valoare unica denumire_prod);
- Toate attributele non-cheie depind in totalitate de toate cheile candidat(ex. pentru tabela produse situatia care ar fi dus la nerespectarea conditiiei pentru a doua forma a fost rezolvata din modelul logic. Astfel, in loc ca pentru fiecare cheie primara sa se repete de acelasi atribut in coloana producator, s-a mai creat o tabela producator care sa contina in mod unic denumirea producatorului)
- Toate attributele non-cheie sunt direct(non-tranzitiv) dependente de toate cheile candidat.(ex. in tabela categorie si subcategorie <Id_Cat> -> Denumire_Cat, <Id_Sub> -> Denumire_Sub si <Id_Sub> -> Id_Cat).



- Proiectare aplicatie Front-End:

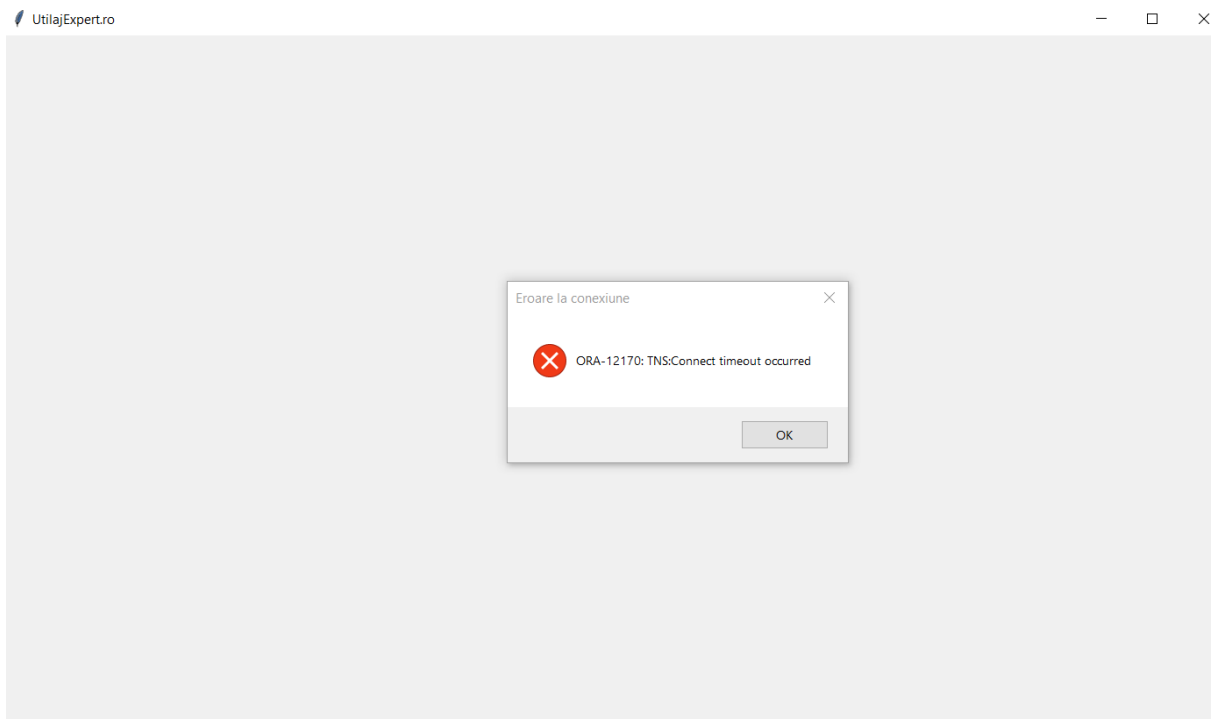
Interfata a fost contruita folosind modulele tkinter pentru interfata si cx_Oracle pentru legarea interfetei de baza de date.

Aplicatia se conecteaza prin modulul cx_Oracle, insa are nevoie de un fisier de dependente numit „instantclient_21_7”, care nu a fost atasat in arhiva proiectului din cauza dimensiunii.

Conectarea are loc la pornirea aplicatiei in clasa panoului principal. Daca conectarea esueaza se afiseaza un text box de eroare.

```
##Conexiune la baza de date
try:
    self.dsn_tns = cx_Oracle.makedsn('bd-dc.cs.tuiasi.ro', '1539', service_name='orcl')
    self.con = cx_Oracle.connect('bd094', 'csbdtip', dsn=self.dsn_tns)
    self.c = self.con.cursor()

except Exception as e:
    messagebox.showerror(title="Eroare la conexiune", message=e)
```



- Structura interfetei:

- Pagina principala: contine un tabel in care sunt afisate din baza de date toate produsele. Ca optiuni avem afisarea tuturor proprietatilor, inchidere, salvare, stergere, editare produs selectat si adaugare nou produs

UtilajExpert.ro

Pagina Principala						
ID	Denumire Produs	Producator	Stoc	Pret	Categorie	Subcategorie
5	Varf de centrare rotativ MK 1	Optimum	24	216.34	accesorii	metal
1	Strung Metal OPTiturn 1503V	Optimum	79	6239.0	metal	strunguri metal
6	Sac colector exhaustor SAA 902	Holzstar	456	9.27	accesorii	lemn
3	Exhaustor rumegus SAA 902	Holzstar	15	1207.02	lemn	exhaustore rumegus
22	hello good	Metallkraft	40	214.0	generatoare curent	generatoare
2	Masina de gaurit tevi RB 127	Metallkraft	24	5851.0	metal	masini de gaurit
4	Fierastrau circular FTS18001	Fixtec	40	1146.6	lemn	fierastraie circulare
7	Generator PG I 12 SR	Unicraft	14	2443.19	generatoare curent	generatoare

Inchide Salveaza Vizualizeaza Sterge

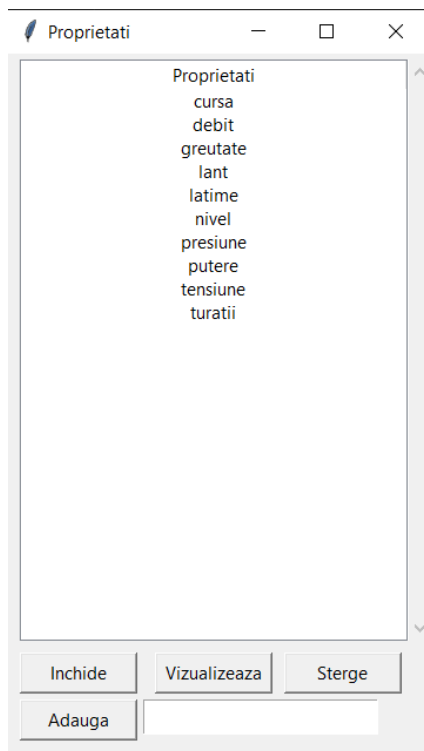
Funcția de stergere a unui produs selectat din tabel

```
def delete(self):
    #TODO sterge linia selectata-done
    selection = self.tree.focus()
    v1= self.tree.item(selection)
    val = v1['values']
    if val == '':
        messagebox.showerror(title='Eroare', message='Niciun produs selectat')
    else:
        m= messagebox.askyesnocancel(title='Warning', message='Esti sigur ca vrei sa stergi produsul?')
        if m == True:
            query = 'delete from produse where denumire_prod = :bind'
            query1 = 'delete from prod_prop where id_prod =:bind'
            query2 = 'select id_prod from produse where denumire_prod = :bind'

            self.c.execute(query2, bind = val[1])
            rows = self.c.fetchall()
            id_prod=None
            for row in rows:
                id_prod = row[0]
            self.c.execute(query1, bind = id_prod)
            self.c.execute(query, bind= val[1])
            self.tree.delete(selection)
```


Funcția de preluare de valori din baza de date și punerea lor în interfață

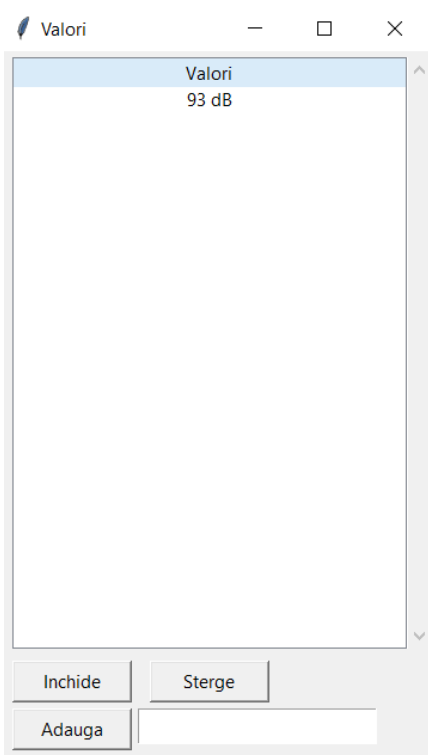
```
def complete_mainTable(self):
    self.c.execute("select p.id_prod, p.denumire_prod, pd.denumire_producator, p.stoc, p.pret, cat.denumire_cat, subcat.denumire_sub\
    \n from produse p, producator pd, categorie cat, subcategorie subcat\
    \n where p.id_producator = pd.id_producator\
    \n and p.id_sub = subcat.id_sub\
    \n and subcat.id_cat = cat.id_cat")
    res = self.c.fetchall()
    for row in res:
        self.tree.insert(parent='', index='end', text='', values=(row[0], row[1], row[2], row[3], row[4], row[5], row[6]))
```



- Pagina proprietati: contine un tabel de o coloana cu denumirile din tabela proprietati. Optiuni: Inchidere, adaugare, vizualizare valori pentru proprietate și stergere.

Funcția de adaugare de proprietati:

```
def add(self):
    #TODO se preia text din self.entry_add si se adauga in tabela-done
    query = 'insert into proprietati(denumire_prop) values (:bind)'
    try:
        self.c.execute(query, bind=self.entry_add.get())
        self.tree.insert(parent='', index='end', text='', values=((self.entry_add.get(),)))
    except Exception as e:
        messagebox.showerror(title='Eroare', message=e)
```



- Pagina Valori: Contine valorile corespunzatoare unei proprietati. Optiuni: Inchidere, stergere, adaugare.

- Pagina Adaugare Produs si Editare Produs este de fapt una si aceeaasi cu diferenta ca pagina de editare vine deja completate cu valorile produsului selectat.

Vizualizare Produs

Denumire Produs
Pret

Producator
Stoc

Descriere
Poza

Categorie
Subcategorie

Adauga Proprietate

Proprietati
cursa pinola
debit aer
greutate
lant
latime sac
nivel zgomot
presiune negativa maxima
putere motor s1
tensiune alimentare
turatii

Inchide

Valori

Salveaza

Vizualizare Produs

Denumire Produs
Pret

Producator
Stoc

Descriere
Poza

Categorie
Subcategorie

Adauga Proprietate

Proprietati
cursa pinola
debit aer
greutate
lant
latime sac
nivel zgomot
presiune negativa maxima
putere motor s1
tensiune alimentare
turatii

Inchide

Valori

Salveaza

In adaugare produs se creeaza un nou produs care este inserat in baza de date, iar in cazul modificarii valorile produsului vechi sunt schimbate direct in baza de date.

