UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA: CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

RaresElectronics

TEMA DE CASĂ
BAZE DE DATE

Coordonator științific
Ing. Sorin AVRAM

Student
Rareș-Andrei DANCĂU
1308A

Iași, 2022

Descrierea proiectului

Proiectul implementează o bază de date pentru un magazin ce comercializează componente electronice. În această bază de date se pot stoca informații despre starea unui produs, numele producătorului și tipul produsului. Combinația acestor caracteristici determină un produs concret, ce are o cantitate disponibilă în magazia deținută de comerciant, și un preț cu care acesta îl va vinde clienților. Mai pot fi înregistrate vânzări, corespunzătoare acestor produse concrete, ce conțin o cantitate dorită de componente și data efectuării vânzării.

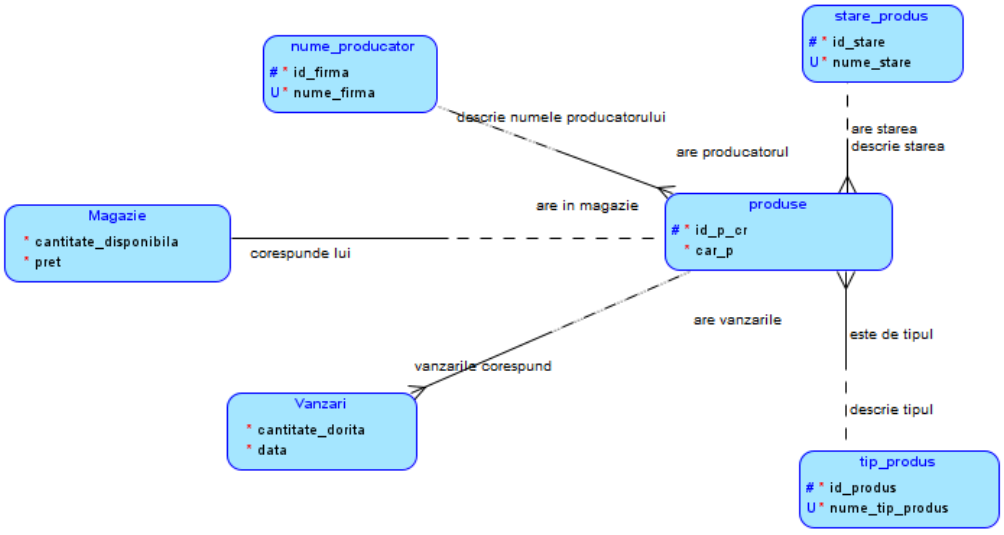
Se cere ca numele stărilor și produselor să nu conțină cifre, să fie unice, toate valorile să fie completate și cantitățile dorite sau disponibile împreună cu prețul să fie pozitive.

Tehnologii folosite

-back-end: Oracle Database Express Edition (XE) Release 11.2.0.2.0
-front-end: Tkinter și cx_Oracle în cadrul Python

Structura și inter-relaționarea tabelelor

Model logic



Relații

1:1: între magazie și produse. Un produs are un corespondent în magazie și ce se află în magazie are un corespondent produs.
1:m: între nume_producător și produse, un nume de producător poate corespunde mai multor produse și mai multe produse pot corespunde unui singur producător. Între tip_produs și produse, pot exista mai multe produse cu același tip de produs și un tip de produs poate corespunde mai multor produse. Între stare_produs și produse, mai multe produse pot avea o

stare și o stare corespunde mai multor produse. Între produse și vânzări, unui produs îi pot corespunde mai multe vânzări, iar mai multor vânzări le pot corespunde același produs.

Normalizare

Tabela produse este în a cincea formă normală deoarece toate atributele sunt valori atomice din domeniul lor, nu există grupuri ce se repetă, celelalte atribute depind direct și total de id-ul produsului concret, îndeplinește afirmația corespunzătoare Boyce-Codd, îndeplinește afirmația celei de a patra forme normale și nu mai poate fi descompusă în sub-tabele cu chei diferite.

Tabela magazie este în a cincea formă normală deoarece toate atributele sunt valori atomice din domeniul lor, nu există grupuri ce se repetă, cantitatea și prețul depind direct și total de id, îndeplinește afirmația corespunzătoare Boyce-Codd, îndeplinește afirmația celei de a patra forme normale și nu mai poate fi descompusă în sub-tabele cu chei diferite.

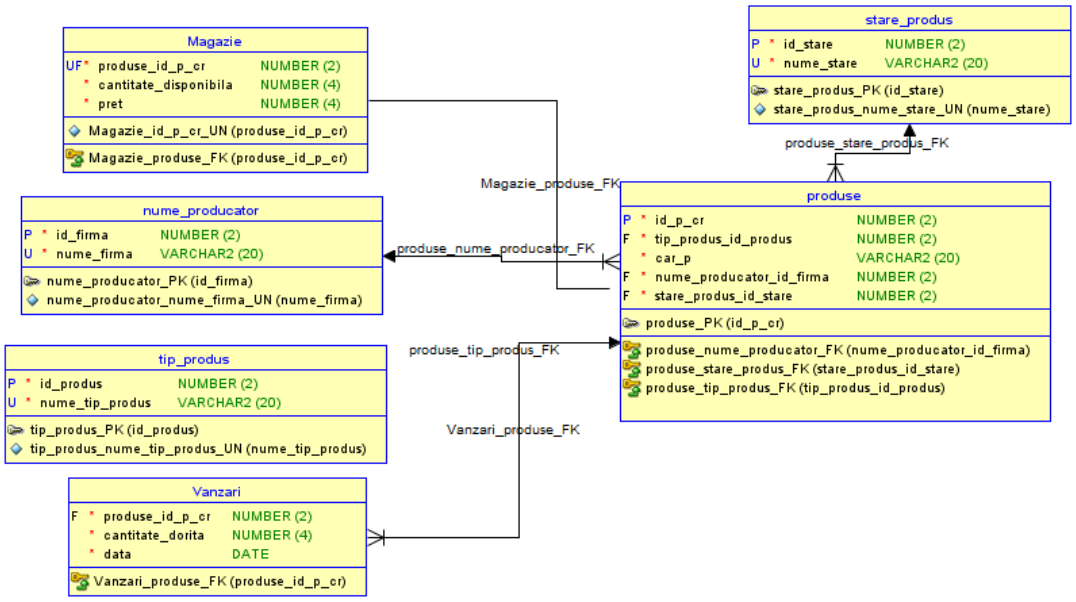
Tabela vânzări nu este în nicio formă canonică deoarece pot exista grupuri ce se repetă.

Tabela nume_producător este în a cincea formă normală deoarece toate atributele sunt valori atomice din domeniul lor, nu există grupuri ce se repetă, numele producătorului depinde direct și total de id, îndeplinește afirmația corespunzătoare Boyce-Codd, îndeplinește afirmația celei de a patra forme normale și nu mai poate fi descompusă în sub-tabele cu chei diferite.

Tabela stare_produs este în a cincea formă normală deoarece toate atributele sunt valori atomice din domeniul lor, nu există grupuri ce se repetă, numele stării depinde direct și total de id, îndeplinește afirmația corespunzătoare Boyce-Codd, îndeplinește afirmația celei de a patra forme normale și nu mai poate fi descompusă în sub-tabele cu chei diferite.

Tabela tip_produs este în a cincea formă normală deoarece toate atributele sunt valori atomice din domeniul lor, nu există grupuri ce se repetă, tipul de produs depinde direct și total de id, îndeplinește afirmația corespunzătoare Boyce-Codd, îndeplinește afirmația celei de a patra forme normale și nu mai poate fi descompusă în sub-tabele cu chei diferite.

Model relațional



Descrierea constrângerilor folosite și de ce au fost acestea necesare

În tabela magazie au fost folosite constrângeri de lungime a câmpurilor pentru a limita lungimea valorilor introduse, o constrângere de unicitate pentru id-ul produsului concret, pentru a nu exista două cantități ale aceluiași produs și constrângeri de not null pentru a nu lăsa date necompletate. S-a mai folosit o constrângere de tip check pentru a verifica faptul că prețul și cantitatea disponibilă unui produs sunt numere pozitive, pentru a se verifica că sunt valori valide. De asemenea, mai este folosită o constrângere de tip foreign key pentru a face legătura cu tabela produse.

În tabela nume_producator sunt constrângeri de lungime, cu aceeași explicație ca la cealaltă tabelă și constrângeri de not null pentru a nu lăsa date necompletate. Mai avem o constrângere de tip cheie primară ce face legătura cu tabela produse.

În tabela produse sunt constrângeri de lungime, constrângeri de not null pentru a nu lăsa date necompletate și una de tip primary key, pentru a face legătura cu tabelele magazie și vânzări. Mai sunt folosite mai multe constrângeri de tip primary key pentru a face legătura cu tabelele stare_produs, tip_produs și nume_producător.

În tabela stare_produs sunt constrângeri de lungime, de tip check pentru a valida numele unei stări, nefiind permise prezența cifrelor în numele stărilor. De asemenea, mai sunt constrângeri de tip primary key pentru id_stare, pentru a face legătura cu tabela produse, și o constrângere unique pentru numele stării, fiind invalide stări cu id diferit, dar același nume.

În tabela tip_produs sunt constrângeri de lungime, de tip check care nu permit ca tipul de produs să conțină cifre, de tip primary key pentru a face legătura cu tabela produse, constrângeri de not null pentru a nu lăsa date necompletate și de unicitate pentru numele produsului.

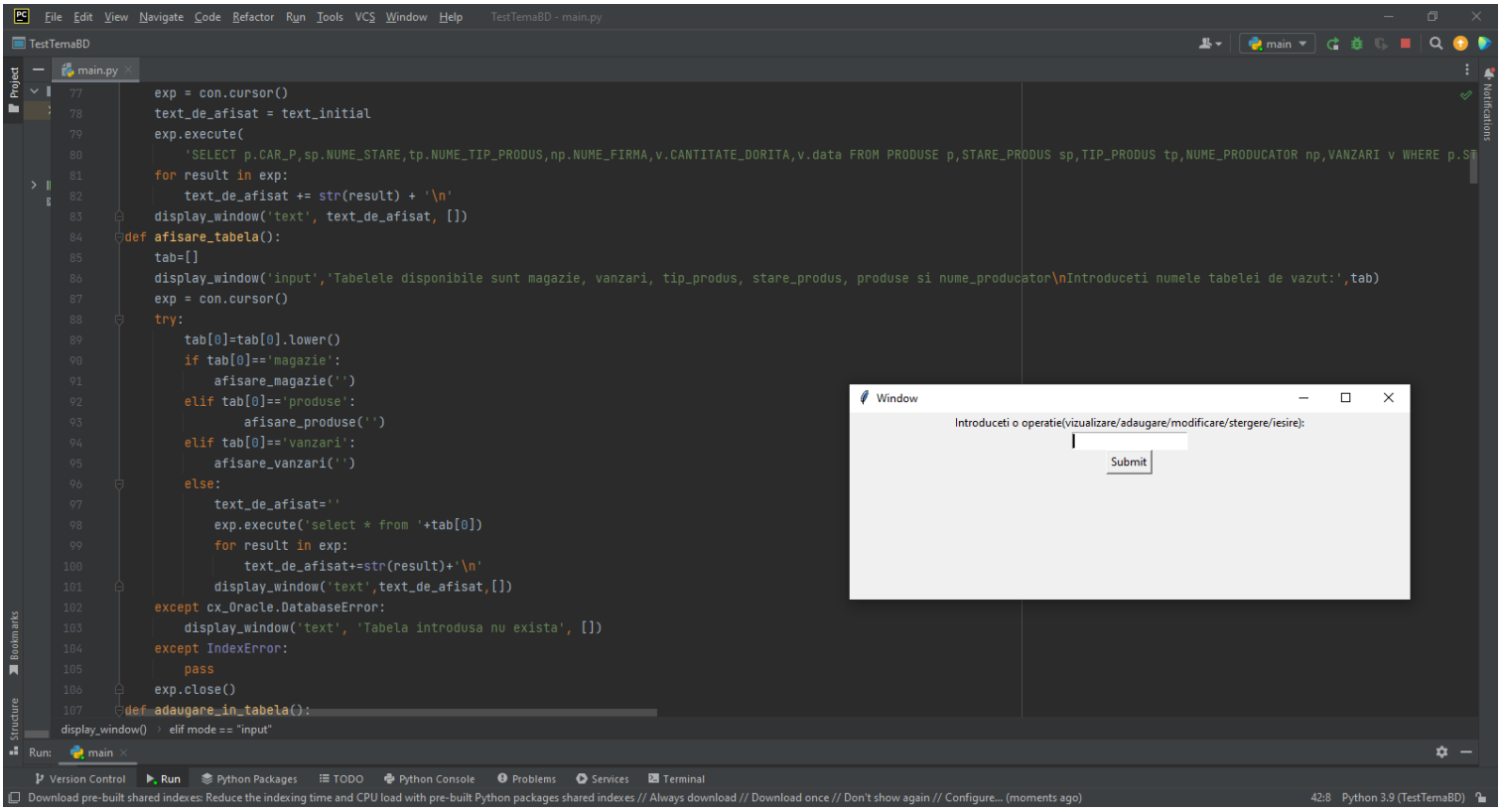
În tabela vanzari sunt constrângeri pentru lungimea parametrilor, constrângeri de not null pentru a nu lăsa date necompletate, validare a cantității dorite și de tip foreign key pentru a face legătura cu tabela produse.

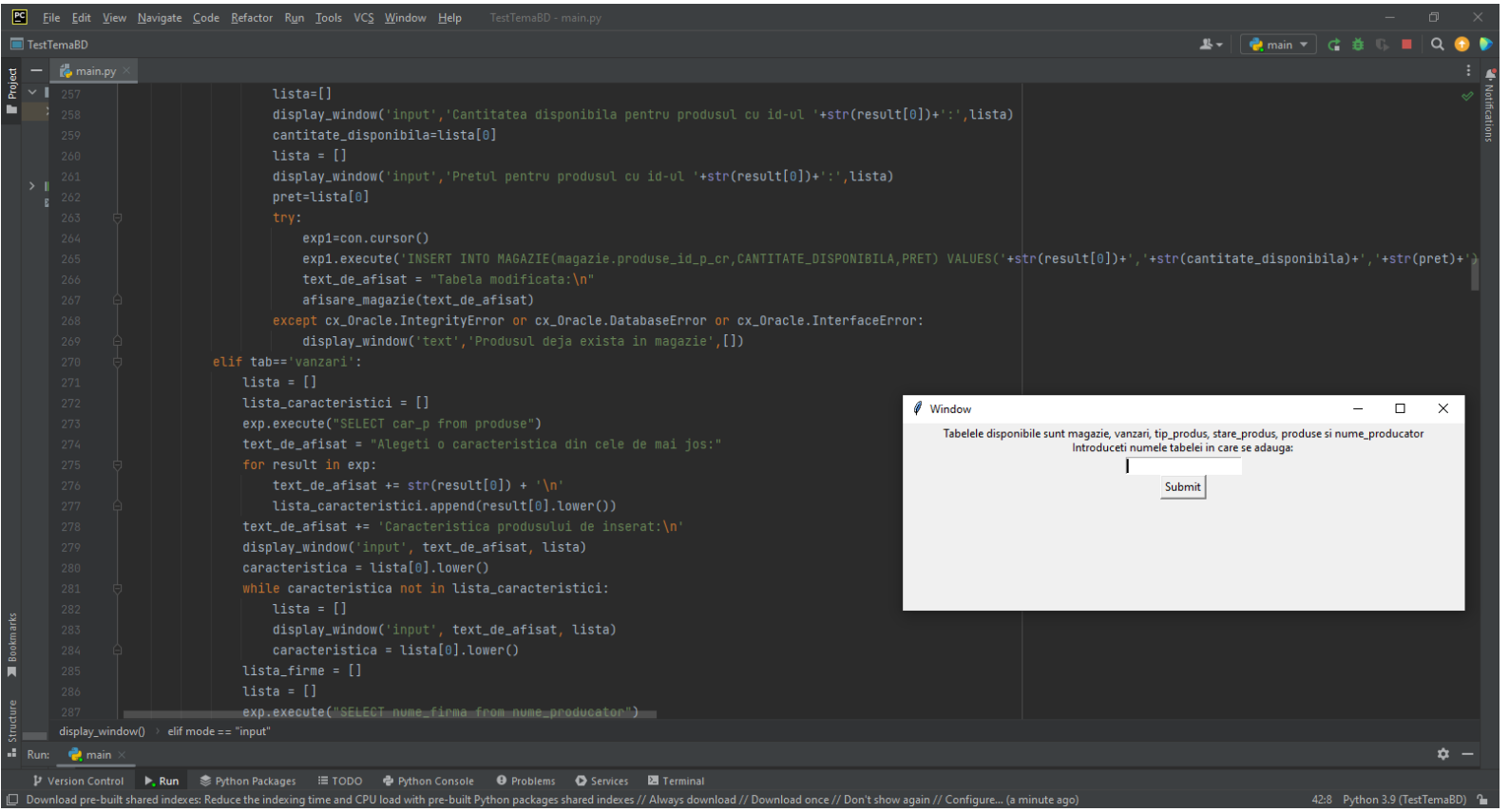
Tranzacția

Întâi se adaugă produsul în tabela produse, după aceea se adaugă în magazie. Când se va adăuga/modifica un rând în tabela vânzări, se va verifica dacă stocul din magazie este suficient pentru a se face adăugarea/modificarea și în caz afirmativ, se va produce inserarea în tabela vânzări și se va actualiza cantitatea din magazie. În cazul în care cantitatea rămasă în magazie nu este suficientă, inserarea în tabela vânzări nu va avea loc și nici actualizarea cantității disponibile din magazie.

Conectarea cu baza de date

Conectarea la baza de date are loc prin obiectul con din aplicație, ce este un obiect de tip cx_Oracle.connect(„nume_utilizator”,”parola”,”detalii_despre_server”). Execuția comenzilor SQL are loc prin obiecte de tip cursor la con.




File Edit View Navigate Code Refactor Run Tools VCS Window Help TestTemaBD - main.py
TestTemaBD
main
main.py
lista=[]
display_window('input','Cantitatea disponibila pentru produsul cu id-ul '+str(result[0])+':',lista)
cantitate_disponibila=lista[0]
lista = []
display_window('input','Pretul pentru produsul cu id-ul '+str(result[0])+':',lista)
pret=lista[0]
try:
exp1=con.cursor()
exp1.execute('INSERT INTO MAGAZIE(magazie.produse_id_p_cr,CANTITATE_DISPONIBILA,PRET) VALUES('+str(result[0])+','+str(cantitate_disponibila)+','+str(pret)+')
text_de_afisat = "Tabela modificata:\n"
afisare_magazie(text_de_afisat)
except cx_Oracle.IntegrityError or cx_Oracle.DatabaseError or cx_Oracle.InterfaceError:
display_window('text','Produsul deja exista in magazie',[])
elif tab=='vanzari':
lista = []
lista_caracteristici = []
exp.execute("SELECT car_p from produse")
text_de_afisat = "Alegeti o caracteristica din cele de mai jos:"
for result in exp:
text_de_afisat += str(result[0]) + '\n'
lista_caracteristici.append(result[0].lower())
text_de_afisat += 'Caracteristica produsului de inserat:\n'
display_window('input', text_de_afisat, lista)
caracteristica = lista[0].lower()
while caracteristica not in lista_caracteristici:
lista = []
display_window('input', text_de_afisat, lista)
caracteristica = lista[0].lower()
lista_firme = []
lista = []
exp.execute("SELECT nume_firma from nume_producator")
Window
Tabelele disponibile sunt magazie, vanzari, tip_produs, stare_produs, produse si nume_producator
Introduceti numele tabelei in care se adauga:
Submit
display_window() elif mode == "input"
Run: main
Version Control Run Python Packages TODO Python Console Problems Services Terminal
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (a minute ago)
42:8 Python 3.9 (TestTemaBD)