



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

DISCIPLINA: BAZE DE DATE

**Tema proiectului:
Gestiunea unui magazin de discuri de vinil
Decembrie 2023**

Student: Colțuneac Iuliana-Diana

Tema proiectului: Gestiunea unui magazin de discuri de vinil

Descrierea proiectului: Un magazin local ce deține o colecție mare de discuri de vinil are nevoie de un sistem de gestiune pentru eficientizarea procesului de vânzare-cumpărare a acestor articole. Discurile pot fi atât noi, cât și utilizate, ca urmare discuri ce conțin același album pot avea starea și prețul diferite. Pentru anumite produse, într-o anumită perioadă, vor exista reduceri. Se dorește, de asemenea, managerierea modului în care produsele sunt stocate și organizate în magazin pentru a facilita accesul și găsirea acestora. Încasările efectuate vor fi adăugate în tratarea problemei pentru a putea vizualiza vânzările și a putea obține date despre acestea (data încasării, metoda de plată).

Proiectul nu tratează problema de administrare a capitalului magazinului, modul în care transportul furnizor-magazin este efectuat sau cumpărarea în format online a discurilor.

În urma analizei cerințelor rezultă:

1. Entitatea vinil_generic

- id_vinilgeneric (cheie primară) - valoare numerică, obligatorie, unică;
- titlu_album - valoare sir de caractere, obligatorie;
- artist - valoare sir de caractere, obligatorie;
- gen_muzical - valoare sir de caractere, obligatorie;
- an_lansare - valoare numerică, optional;

1FN: Da, deoarece toate atributele conțin valori atomice.

2FN: Da, deoarece toate atributele depind de cheia primară id_vinilgeneric.

3FN: Da, deoarece nu există dependențe funcționale tranzitive. Atributele necheie (titlu_album, artist, gen_muzical, an_lansare) nu depind de alte atribute necheie.

2. Entitatea vinil_fizic

- id_vinil (cheie primară) - valoare numerică, obligatorie, unică;
- stare (bună, foarte bună, nou) - valoare sir de caractere, obligatoriu;
- preț - valoare numerică, obligatorie;
- stoc - valoare numerică, obligatorie;
- id_furnizor - valoare numerica, obligatorie;
- id_promoție - valoare numerică, obligatorie;

1FN: Da, deoarece toate atributele conțin valori atomice.

2FN: Da, deoarece nu există dependențe funcționale parțiale. Toate atributele depind de cheia primară id_vinil.

3FN: Da, deoarece nu există dependențe funcționale tranzitive. Atributele necheie (stare, preț, stoc) nu depind de alte atribute necheie.

3. Entitatea promoție

- id (cheie primară) - valoare numerică, obligatorie, unică;
- data_inc - valoare de tip data, obligatorie;
- data_sf - valoare de tip data, obligatorie;
- procent - valoare numerică, obligatorie;

1FN: Da, deoarece toate atributele conțin valori atomice.

2FN: Da, deoarece toate atributele depind de cheia primară id_promotie.

3FN: Da, deoarece nu există dependențe funcționale tranzitive. Atributele necheie (data_inc, data_sf, procent) nu depind de alte atribute necheie.

4. Entitatea poziționare

-biblioteca - valoare numerică, obligatorie;

-raft - valoare numerică;

-id_vinil - valoare numerică, obligatorie, unică;

FN: Da, deoarece toate atributele conțin valori atomice și nu există grupuri repetate de coloane.

2FN: Da, deoarece nu există dependențe funcționale parțiale. Toate atributele depind de cheia primară compusă biblioteca și raft.

3FN: Da, deoarece nu există dependențe funcționale tranzitive. Atributele necheie (biblioteca, raft) nu depind de alte atribute necheie.

5. Entitatea încasări

-data_vânzare - valoare de tip data;

-metoda_de_plata(cash/card) - valoare sir de caractere, obligatoriu;

-suma - valoare numerică, obligatorie;

-id_vinil - valoare numerică, obligatorie, unică;

1FN: Da, deoarece toate atributele conțin valori atomice.

2FN: Da, deoarece nu există dependențe funcționale parțiale. Toate atributele depind de cheia primară id_incasari.

3FN: Da, deoarece nu există dependențe funcționale tranzitive. Atributele necheie (data_vânzare, metoda_de_plata, suma) nu depind de alte atribute necheie.

6. Entitatea furnizor

-id(cheie primară) - valoare numerică, obligatorie, unică;

-denumire - valoare sir de caractere, obligatorie;

-adresa - valoare sir de caractere

1FN: Da, deoarece toate atributele conțin valori atomice și nu există grupuri repetate de coloane.

2FN: Da, deoarece nu există dependențe funcționale parțiale. Toate atributele depind de cheia primară id_furnizor.

3FN: Da, deoarece nu există dependențe funcționale tranzitive. Atributele necheie (denumire, adresa) nu depind de alte atribute necheie.

În proiectarea acestei baze de date s-au identificat următoarele tipuri de relații:

1:1 (one-to-one), 1:n (one-to-many)

Între tabela **vinil_generic** și tabela **vinil_fizic** se stabilește o relație de 1:n. Un disc generic poate avea un stoc de viniluri fizice ce diferă prin caracteristici, astfel, entitatea vinil_fizic implementează entitatea vinil_generic, legătura dintre cele două făcându-se prin câmpul id_vinil_generic;

Între tabela **vinil_fizic** și tabela **pozitionare** se stabilește o relație de 1:1. Poziția unui set de discuri fizice cu aceleași caracteristici se poate verifica prin vizualizarea câmpurilor tabelii pozitionare. Un set de discuri se poate afla într-o singură poziție la un moment dat, iar într-o

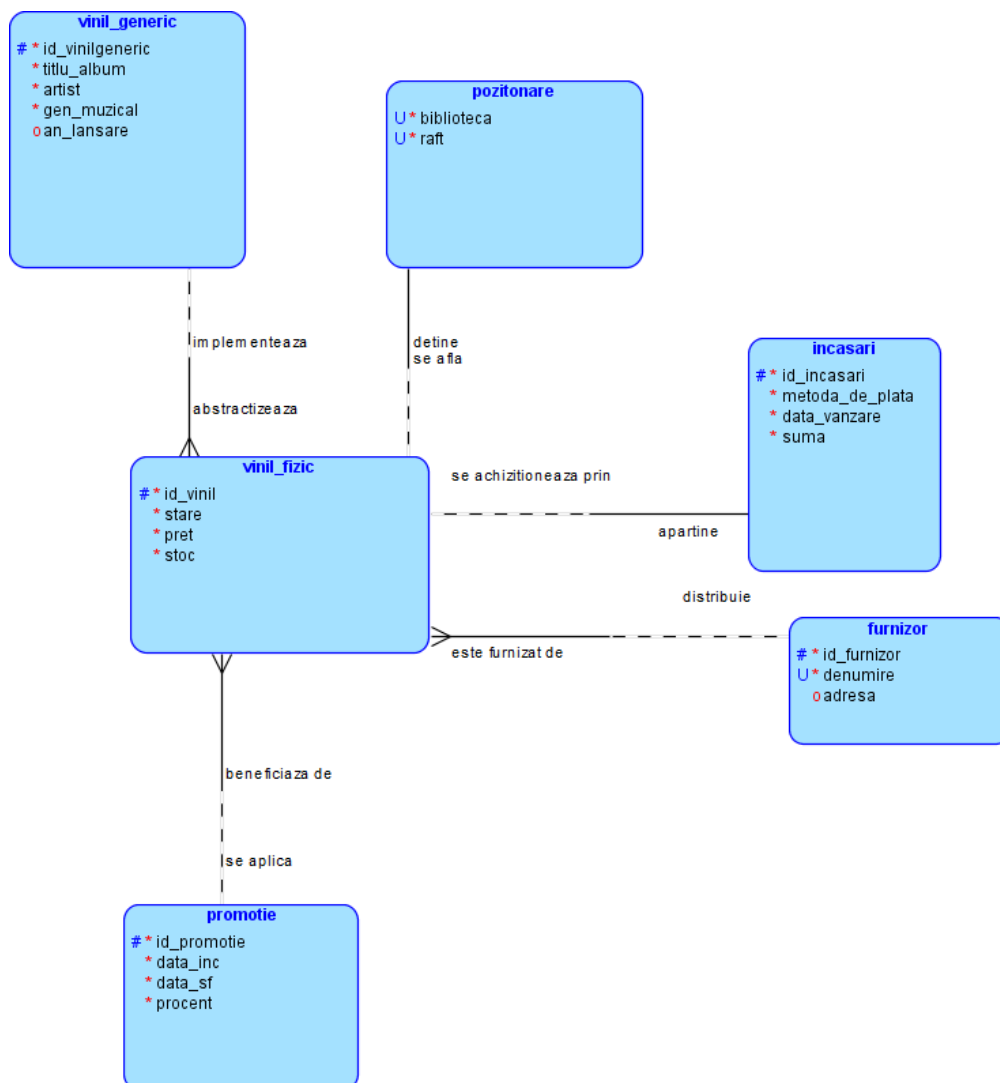
anumită bibliotecă, pe un anumit raft se poate afla un singur set de dicuri. Relația dintre cele două se stabilește prin coloana id_vinil;

Între tabela **vinil_fizic** și tabela **incasari** se stabilește o relație de 1:1. Un disc poate fi vândut printr-o singură încasare. Relația dintre cele două se stabilește prin coloana id_vinil;

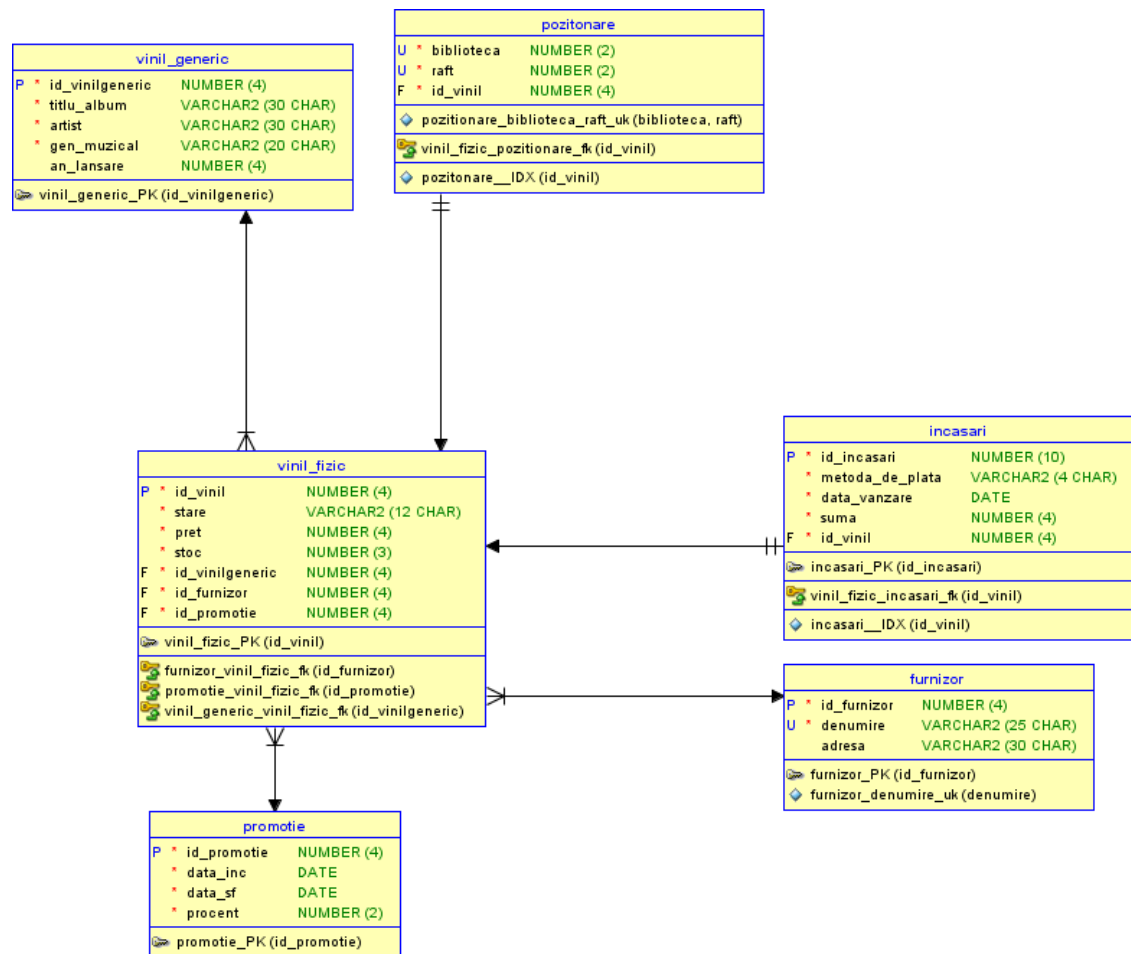
Între tabela **vinil_fizic** și tabela **furnizor** se stabilește o relație de 1:n. Un furnizor poate distribui mai multe discuri pentru magazin, dar un disc poate fi furnizat de un singur distribuitor. Relația dintre cele două se stabilește prin coloana id_furnizor;

Între tabela **promoție** și tabela **vinil_fizic** se stabilește o relație de 1:n. O promoție se poate aplica mai multor discuri, dar un disc poate beneficia de o singură promoție la un moment dat. Relația dintre cele două se stabilește prin intermediul câmpului id_promotie;

Modelul logic:



Modelul relațional:



Descrierea constrângerilor:

În cadrul proiectării bazei de date s-au găsit necesare următoarele constrângeri:

Vinil_generic:

1. Constrângere de tip **CHECK** privitoare la lungimea denumirii albumului:
(length(titlu_album) > 0)
2. Constrângere de tip **CHECK** privitoare la lungimea numelui artistului:
(length(artist) > 1)
3. Constrângere de tip **CHECK** privitoare la lungimea denumirii genului muzical în care se încadrează albumul:
(length(gen_muzical) > 1)
4. Constrângere de tip **PRIMARY KEY** (id_vinilgeneric);

Vinil_fizic:

5. Constrângere de tip **CHECK** privitoare la prețul unui disc:
(pret > 0)
6. Constrângere de tip **CHECK** privitoare la stocul unui set de discuri:
(stoc >= 0)
7. Constrângere de tip **PRIMARY KEY (id_vinil);**
8. Constrângere de tip **CHECK** privitoare la starea unui disc:
CHECK (stare IN ('buna', 'foarte buna', 'noua'))

Promotie:

9. Constrângere de tip **CHECK** privitoare la perioada unei promoții. Aceasta trebuie să fie valabilă cel puțin o săptămână:
(data_sf - data_inc > 7)
10. Constrângere de tip **CHECK** privitoare la valoarea procentajului promoției. Pentru ca o promoție să fie validă aceasta trebuie să nu fie nulă, dar nici să depășească prețul produsului:
(procent > 0 AND procent < 100)
11. Constrângere de tip **PRIMARY KEY (id_promotie);**

Poziționare:

12. Constrângere de tip **CHECK** privitoare la numărul de ordine al bibliotecii în care se află un disc:
(biblioteca > 0)
13. Constrângere de tip **CHECK** privitoare la numărul de ordine al raftului pe care se așează un set de discuri:
(raft > 0 OR raft BETWEEN 1 AND 15)

* condițiile referitoare la raft și bibliotecă s-au stabilit în acest mod deoarece numărul de biblioteci deținute de magazin se poate mări oricât de mult, pe când numărul de rafturi rămâne constant;

14. Constrângere de tip **UNIQUE (biblioteca, raft):**
*spațiul dintr-o anumită bibliotecă, mărginit de un anumit raft poate depozita un singur set de discuri;

Furnizor:

15. Constrângere de tip **CHECK** referitoare la denumirea furnizorului:
length(denumire) > 1
16. Constrângere de tip **PRIMARY KEY (id_furnizor);**
17. Constrângere de tip **UNIQUE (denumire):**
*firmele de distribuire trebuie să fie definite printr-un nume unic;

Încasări:

18. Constrângere de tip **CHECK** referitoare la metoda prin care s-a efectuat plata:
CHECK (metoda_de_plata IN ('card', 'cash'))

19. Constrângere de tip **CHECK** referitoare la suma intrată în posesia magazinului (orice încasare trebuie să depășească valoarea 0)
CHECK (suma > 0)
20. Constrângere de tip **PRIMARY KEY (id_incasari);**

Pe lângă acestea, s-au adăugat și constrângeri de tip cheie străină pentru a face posibilă comunicarea dintre următoarele tabele:

21. Vinil_fizic – furnizor:
FOREIGN KEY (id_furnizor)
REFERENCES furnizor (id_furnizor);
22. Vinil_fizic – furnizor:
FOREIGN KEY (id_promotie)
REFERENCES promotie (id_promotie);
23. Incasari – Vinil_fizic
FOREIGN KEY (id_vinil)
REFERENCES vinil_fizic (id_vinil);
24. Pozitionare – Vinil_fizic
FOREIGN KEY (id_vinil)
REFERENCES vinil_fizic (id_vinil);
25. Vinil_fizic – Vinil_generic
FOREIGN KEY (id_vinilgeneric)
REFERENCES vinil_generic (id_vinilgeneric);
26. Constrângerile de tip **NOT NULL** se găsesc pe marea majoritate din atributele din tabele, cu excepția următoarelor două ce nu sunt de o prioritate ridicată:
1. Furnizor-adresa;
 2. Vinil_generic: an_lansare;

**** Primary key-urile sunt generate de baza de date printr-un mecanism de tip autoincrement;**

Pentru realizarea aplicației s-au folosit următoarele tehnologii:

1. **Python versiunea 3.12** care poate fi descărcată de pe site-ul oficial <https://www.python.org/downloads/>.
Aplicația are nevoie de următoarele biblioteci:
 - 1.1. **Click versiunea 8.0.1.**
Click este un pachet Python folosit pentru a crea interfețe în linia de comandă, într-un mod eficient;
 - 1.2. **Flask versiunea 2.1.1.**
Flask este un API (Application Programming Interface) Python ce permite construirea de aplicații web. A fost dezvoltat de Armin Ronacher (un programator austriac și un speaker cunoscut la conferințele în domeniul software).

1.3. cx_Oracle

cx_Oracle este un modul de extensie Python care permite accesul la Oracle Database.

Conformă cu specificația API 2.0 a bazei de date Python, cu un număr considerabil de adăugiri și câteva excluderi.

Pentru a utiliza cx_Oracle cu Python și Oracle Database a fost necesară descărcarea unei biblioteci Client Oracle. Astfel, s-a folosit **instantclient_21_12** de la Oracle Instant Client.

Conectarea propriu-zisă la baza de date s-a realizat astfel:

```
# Establish a connection
con = cx_Oracle.connect(user=username, password=password, dsn=dsn)
```

Funcționalitatea aplicației

Prin aceasta aplicație am implementat vizual operabilitatea principalelor funcții de interogare a unei baze de date. Astfel, putem adauga/modifica/sterge sau vizualiza o baza de date cu ușurință:

Funcția **SELECT**:

```
vinil_generic = []

cur = con.cursor()
cur.execute('select * from vinil_generic')
for result in cur:
    vinil_generic1 = { }
    vinil_generic1['id_vinilgeneric'] = result[0]
    vinil_generic1['titlu_album'] = result[1]
    vinil_generic1['artist'] = result[2]
    vinil_generic1['gen_muzical'] = result[3]
    vinil_generic1['an_lansare'] = result[4]

    vinil_generic.append(vinil_generic1)
cur.close()
print(vinil_generic)
return render_template('vinil_generic.html',vinil_generic=vinil_generic)
```

ID_vinilgeneric	Titlu_Album	Artist	gen_Muzical	An_Lansare	Action
1	Saxophone Colossus	Sonny Rollins	Jazz	1957	<div>Sterge</div> <div>Editeaza</div>
2	Bohemian Rhapsody	Queen	Rock	2006	<div>Sterge</div> <div>Editeaza</div>
3	Pet Sounds	The Beach Boys	Pop	1966	<div>Sterge</div> <div>Editeaza</div>
4	The Masterpieces	Ludwig van Beethoven	Clasic	2007	<div>Sterge</div> <div>Editeaza</div>

Funcția **UPDATE**:

```
emp=0
cur = con.cursor()
titlu_album="" + request.form['titlu_album'] + ""

cur.execute('select id_vinilgeneric from vinil_generic
where titlu_album='+titlu_album)
for result in cur:
    emp=result[0]
cur.close()
artist = "" + request.form['artist'] + ""
gen_muzical = "" + request.form['gen_muzical'] + ""
an_lansare = request.form['an_lansare']
query = "UPDATE vinil_generic SET titlu_album=%s,
artist=%s, gen_muzical=%s, an_lansare=%s
where id_vinilgeneric=%s" %
(titlu_album,artist,gen_muzical,an_lansare, emp)
```

Editeaza Vinil Generic

Titlu_album	<input type="text" value="Saxophone Colossus"/>
Artist	<input type="text"/>
gen muzical	<input type="text" value="Jazz"/>
An_Lansare	<input type="text" value="1957"/>

Funcția **INSERT**:

```
values = []
values.append('"' + str(vinil) + '"')
values.append('"' + request.form['stare'] + '"')
values.append('"' + request.form['pret'] + '"')
values.append('"' + request.form['stoc'] + '"')
values.append('"' + request.form['id_furnizor'] + '"')
values.append('"' + request.form['id_promotie'] + '"')
values.append('"' + request.form['id_vinilgeneric'] + '"')
fields = ['id_vinil','stare', 'pret', 'stoc', 'id_furnizor', 'id_promotie', 'id_vinilgeneric']
query = 'INSERT INTO %s (%s) VALUES (%s)' % ('vinil_fizic', ','.join(fields), ','.join(values))
```

Adauga Vinil Fizic

Stare

☐ Buna ☐ Foarte Buna ☐ Noua

Pret

Stoc

ID Furnizor

ID Promotie

ID Vinil generic

Adauga Vinil Fizic