

Proiect Baze de date 2

Baza de date pentru magazin de muzica online

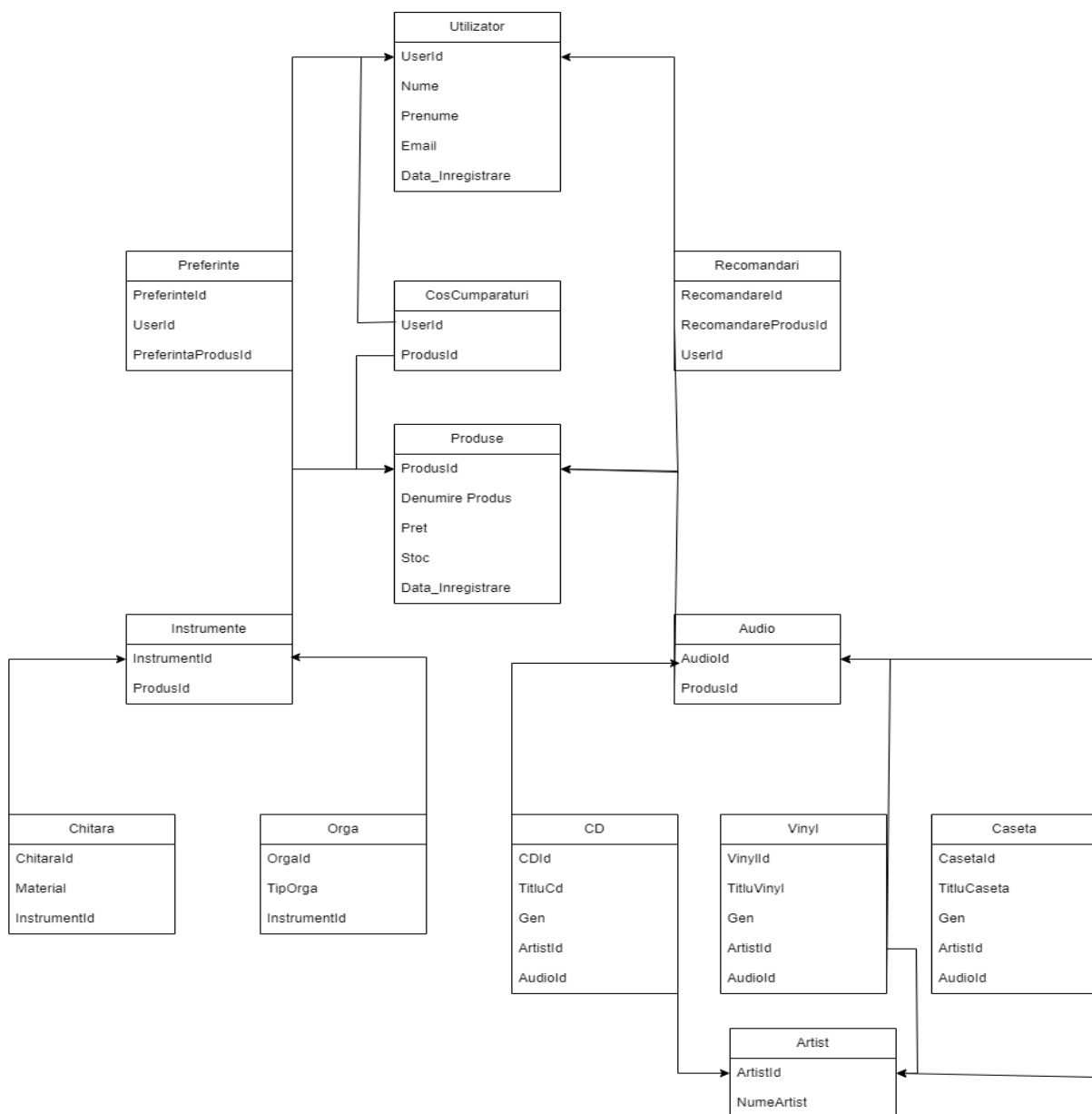
Andronic Dragos Sebastian – 342C3

Facultatea de Automatica si Calculatoare 2024

1. Descrierea temei:

Proiectul consta in implementarea unei baze de date pentru un magazin de muzica online ce contine doua categorii de produse: instrumente (pentru acest magazin online se comercializeaza chirari si orgi) si muzica in diferite formate (CD, vinyl si casete). Un utilizator poate sa isi adauge produse in cosul de cumparaturi, sa adauge anumite produse la o lista de preferinte si sa aiba parte de anumite recomandari de produse.

1.1 Diagrama bazei de date:



1.2 Structura tabelelor:

Tabelele Utilizator si Produse sunt utilizate pentru a retine informatii generale despre un utilizator si un produs. Tabelele CosCumparaturi, Preferinte si Recomandari creeaza asocieri între cele doua tabele (legaturi cu id-ul celor doua valori), retin acelasi tip de informatie dar au semnificatii diferite.

Tabelele Audio si Instrumente fac legatura între tabela Produse si tabelele de pe nivelul 4 (Chitara, Orga, CD, Vinyl si Caseta). In tabela Chitara retin si tipul de material folosit pentru fabricarea unei chitari, iar in tabela Orga retin si tipul de orga.

Tabelele CD, Vinyl si Caseta retin titlul si genul de muzica si sunt legate de tabela Artist.

1.3 Descrierea constrangerilor de integritate:

In crearea tabelelor am adaugat constrangeri NOT NULL pentru valorile informatiile de tipul VARCHAR si INT, ca PRIMARY KEY aproape toate tabele (exceptand cele care sunt doar asociative) utilizeaza Ca FOREIGN KEY am adaugat urmatoarele asocieri:

- CosCumparaturi -> Utilizator
- CosCumparaturi -> Produse
- Recomandari -> Utilizator
- Recomandari -> Produse
- Preferinte -> Utilizator
- Preferinte -> Produse
- Instrumente -> Produse
- Audio -> Produse
- Chitara -> Instrumente
- Orga -> Instrumente
- CD -> Audio
- CD -> Artist
- Vinyl-> Audio
- Vinyl-> Artist
- Caseta -> Audio
- Caseta -> Artist

1.4 Descrierea procedurilor:

Am definit trei proceduri pentru a afisa date relevante.

```
CREATE OR REPLACE PACKAGE pck_aplicatie AS
    PROCEDURE raport_artist_audio_disponibil_toate_formatele(
        out_cursor OUT SYS_REFCURSOR,
        artist_name_param IN VARCHAR2
    );

    PROCEDURE raport_chitara_acelasi_material(
        out_cursor OUT SYS_REFCURSOR,
        user_id_param IN NUMBER
    );

    PROCEDURE raport_recomandare_audio_acelasi_pret(
        out_cursor OUT SYS_REFCURSOR,
        user_id_param IN NUMBER
    );
END pck_aplicatie;
/
```

Raport de complexitate minim 4: (intoarece albumele de acelasi pret cu cele din tabela recomandare)

```
PROCEDURE raport_recomandare_audio_acelasi_pret(  
    out_cursor OUT SYS_REFCURSOR,  
    user_id_param IN NUMBER  
) AS  
BEGIN  
    OPEN out_cursor FOR  
        -- Query 4  
        SELECT  
            CD.CDId,  
            CD.TitluCD,  
            CD.Gen,  
            CD.ArtistId_FK8,  
            CD.AudioId_FK8  
        FROM  
            Recomandari R  
        JOIN  
            Produse P ON R.RecomandareProdusId = P.ProdusId  
        JOIN  
            Audio A ON P.ProdusId = A.ProdusId_FK5  
        JOIN  
            CD ON A.AudioId = CD.AudioId_FK8  
        WHERE  
            R.UserId_FK3 = user_id_param;  
    COMMIT;  
END raport_recomandare_audio_acelasi_pret;
```

Raport de complexitate minim 6: (gaseste o materialul unei chitari din tabela preferinte pentru un utilizator si intoarce celalalte chitari de care sunt de acelasi material)

```
PROCEDURE raport_chitara_acelasi_material(  
    out_cursor OUT SYS_REFCURSOR,  
    user_id_param IN NUMBER  
) AS  
BEGIN  
    OPEN out_cursor FOR  
        -- Query 6  
        SELECT  
            P.ProdusId,  
            P.DenumireProdus,  
            P.Pret,  
            P.Stoc  
        FROM Produse P  
        JOIN Instrumente I ON P.ProdusId = I.ProdusId_FK4  
        JOIN Chitara C ON I.InstrumentId = C.InstrumentId_FK6  
        WHERE  
            C.Material = (SELECT C.Material FROM Chitara C  
                          JOIN Preferinte P ON C.InstrumentId_FK6 = P.PreferintaProdusId  
                          WHERE P.UserId_FK2 = user_id_param)  
            AND P.ProdusId NOT IN (SELECT PreferintaProdusId FROM Preferinte WHERE UserId_FK2 = user_id_param);  
    COMMIT;  
END raport_chitara_acelasi_material;
```

Raport de complexitate minim 7: (intoarce numarul de albume – titluri de album – pentru un artist pentru fiecare tip de format audio)

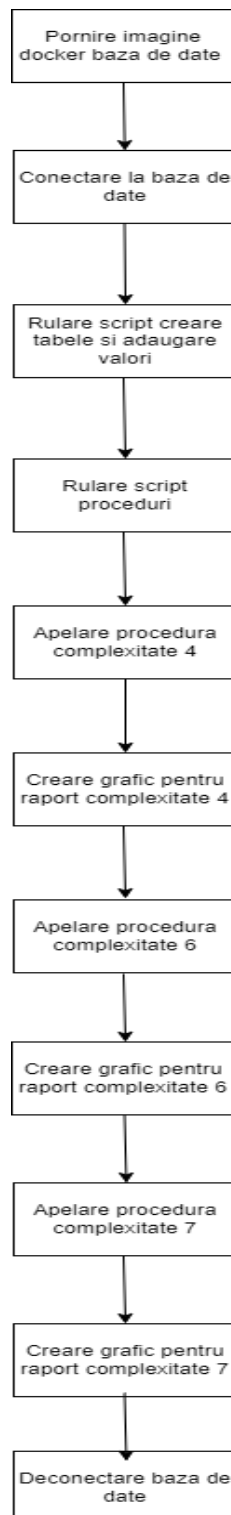
```
CREATE OR REPLACE PACKAGE BODY pck_aplicatie AS  
    PROCEDURE raport_artist_audio_disponibil_toate_formatele(  
        out_cursor OUT SYS_REFCURSOR,  
        artist_name_param IN VARCHAR2  
    ) AS  
    BEGIN  
        OPEN out_cursor FOR  
            SELECT  
                artist_name_param AS Artist,  
                COUNT(DISTINCT C.CasetaId) AS NumarCasetaDisponibile,  
                COUNT(DISTINCT CD.CDId) AS NumarCDDDisponibile,  
                COUNT(DISTINCT V.VinylId) AS NumarVinylDisponibile  
            FROM Artist A  
            LEFT JOIN Caseta C ON A.ArtistId = C.ArtistId_FK10  
            LEFT JOIN CD ON A.ArtistId = CD.ArtistId_FK8  
            LEFT JOIN Vinyl V ON A.ArtistId = V.ArtistId_FK9 AND V.TitluVinyl LIKE '% - Vinyl'  
            WHERE  
                A.NumeArtist = artist_name_param  
                AND C.AudioId_FK10 IN (SELECT ProdusId FROM Produse WHERE Stoc > 0)  
                AND CD.AudioId_FK8 IN (SELECT ProdusId FROM Produse WHERE Stoc > 0)  
                AND V.AudioId_FK9 IN (SELECT ProdusId FROM Produse WHERE Stoc > 0)  
            GROUP BY A.NumeArtist;  
    COMMIT;  
END raport_artist_audio_disponibil_toate_formatele;
```

Relevanta procedurilor: pentru raportul de complexitate 4 este util ca intr-o aplicatie de gestiune pentru un magazin online sa avem cat mai multe cautari dupa criterii si pretul poate reprezenta un criteriu bun, asemenea si materialul unei chitari dintr-o lista de preferinte, cat si disponibilitatea albumelor pentru un artist.

2. Descrierea aplicatiei

Pentru implementare si crearea de grafice am folosit Python. Am create doua scripturi cel de aplicatie (script_app.py) in care ma conectez la baza de date, apelez procedurile stocate si apelez functiile din grafice.py, iar la final inchid conexiunea cu baza de date.

2.1 Diagrama de stari:



2.2 Conectarea la baza de date:

Am folosit metodele din laboratorul 9 de `openConnection` si `closeConnection`. Pentru prima metoda am adaugat si executarea instructiunilor SQL din scriptul de creare a tabelelor si inserarea valorilor.

Conectarea la baza de date se face cu urmatoarele date:

```
if __name__ == "__main__":  
    oc = OracleConnection('localhost', 1521, 'XE', 'system', 'parolaAiaPuternic4')  
    oc.openConnection()
```

Funcția `openConnection`:

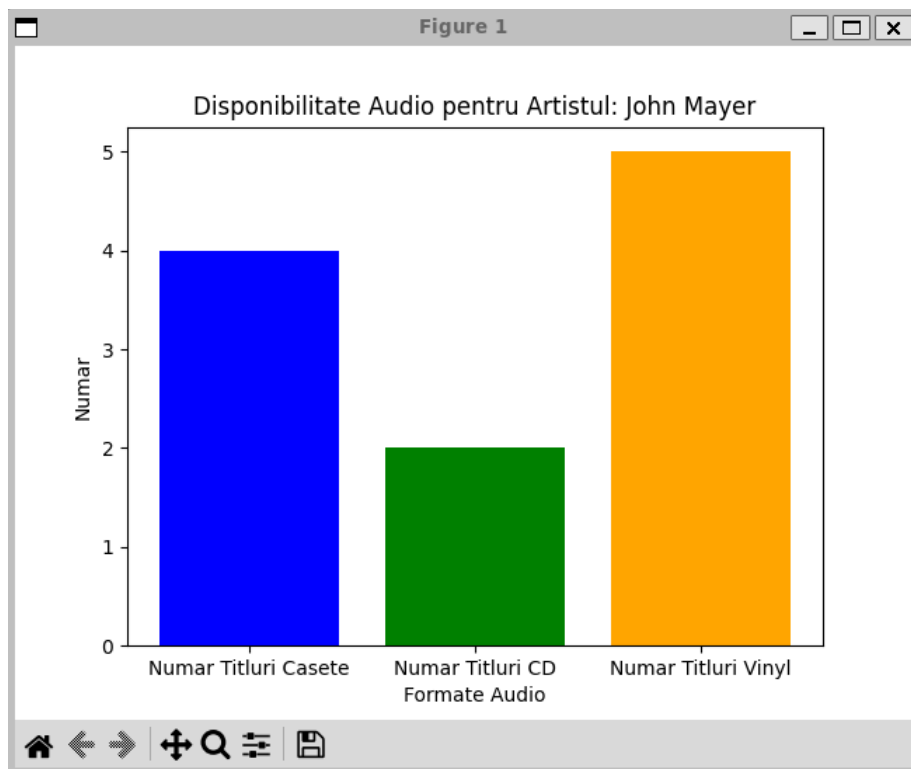
```
def openConnection(self):  
    try:  
        dsn_tns = cx_Oracle.makedsn(self.host, self.port, self.schema)  
        self.db = cx_Oracle.connect(self.username, self.password, dsn_tns)  
        self.cursor = self.db.cursor()  
  
        with open('script_create.sql', 'r') as f:  
            script = f.read()  
  
        # Split the script into individual statements  
        statements = script.split(';')  
  
        # Execute each SQL statement  
        for statement in statements:  
            statement = statement.strip()  
            if statement:  
                try:  
                    self.cursor.execute(statement)  
                    print(f"Statement executed successfully: {statement}")  
                except cx_Oracle.DatabaseError as ex:  
                    error, = ex.args  
                    print(f"Error code: {error.code}, message: {error.message}")  
  
        print("Script execution completed!")  
        print("Connection open!")  
    except Exception as e:  
        print("Connection not open!")  
        print(e)
```

Funcția `closeConnection`:

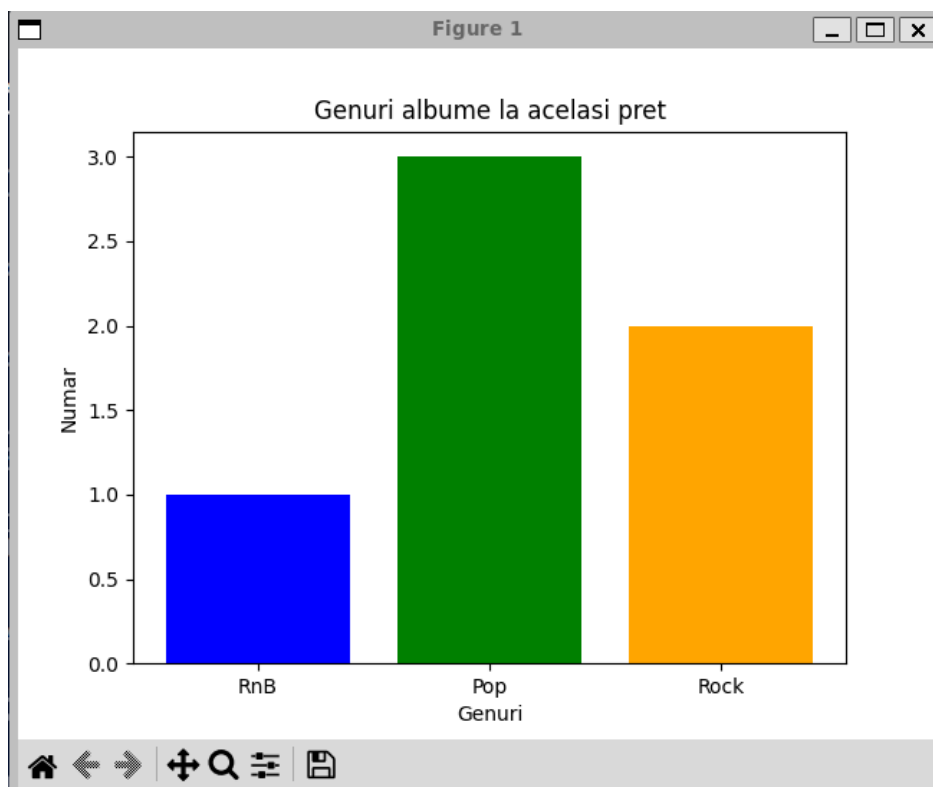
```
def closeConnection(self):  
    try:  
        self.cursor.close()  
        self.db.close()  
        print("Connection close!")  
    except Exception as e:  
        print("Connection not closed!")  
        print(e)
```

3. Capturi ecrane rapoarte

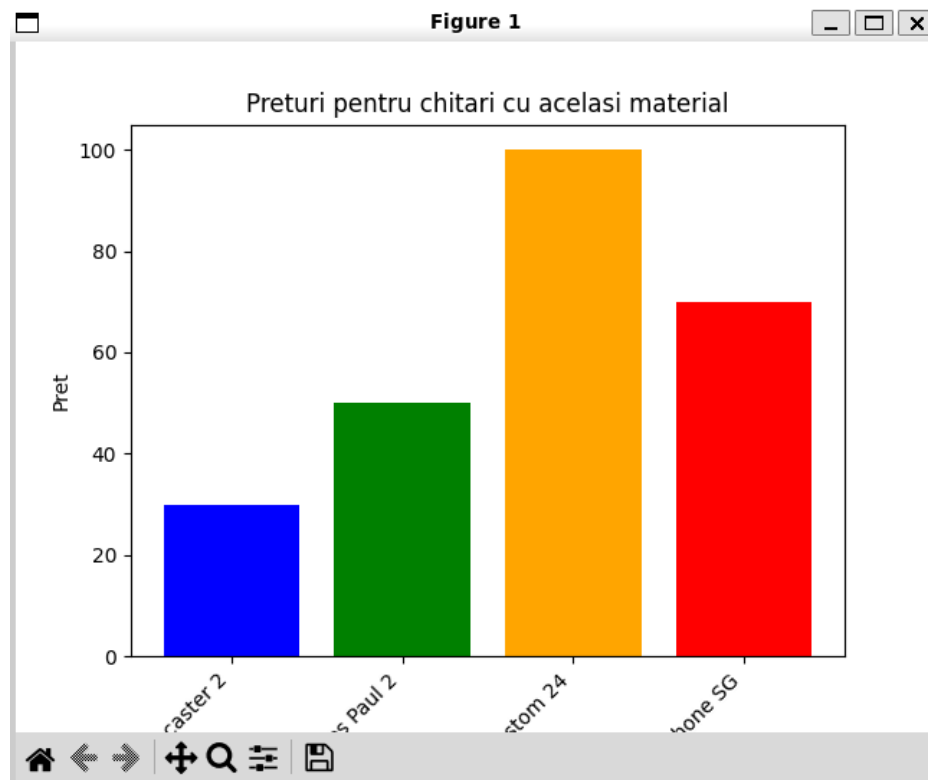
Grafic raport complexitate minim 7:



Grafic raport complexitate minim 4:



Grafic raport complexitate minim 6:



4. Concluzii:

Proiectul actual reprezinta o baza pentru dezvoltarea unei aplicatii adecvate temei magazin online de muzica si consider ca poate fi extins si implementat, consider ca proiectul este scalabil si se pot adauga foarte usor mai multe entitati si noi functionalitati pentru aceasta aplicatie minimala. Am inteles de ce este importanta proiectarea bazei de date cat si intelegerea relatiilor si constrangerilor in momentul proiectarii.

Consider ca a fost un proiect util pentru mine in care am fost pus in situatia de a proiecta o baza de date, de a ma conecta la imaginea de docker si de a realiza proceduri pe implementarea mea. Am consolidat cunostintele legate de matplotlib din Python si cunostintele legate de baze de date in general.

5. Bibliografie:

- 5.1 <https://ocw.cs.pub.ro/courses/bd2/laboratoare/09>
- 5.2 https://odpi-c.readthedocs.io/en/latest/user_guide/installation.html
- 5.3 <https://ocw.cs.pub.ro/courses/bd2/resurse/docker>
- 5.4 <https://ocw.cs.pub.ro/courses/bd2/laboratoare/05>