

Proiect baze de date
~ aplicație pentru muzică / podcast-uri ~

Ingeaua Alexandru-Cristian, FMI, grupa 131

CUPRINS

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.....	3
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.....	3
3. Descrierea entităților, incluzând precizarea cheii primare.....	4
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.....	4
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicate, valori posibile ale atributelor.....	6
6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.....	12
7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.....	13
8. Enumerarea schemelor relationale corespunzătoare diagramei conceptuale proiectate la punctul 7.....	14
9. Realizarea normalizării până la forma normală 3 (FN1-FN3)....	15
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele.....	18
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ și minimum 10 înregistrări în tabelele asociative).....	21
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:.	51
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.....	58
14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.....	62
15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minim 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.....	65

Exercitiul 16 - pagina 71

Exercitiul 17 - pagina 75

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

Modelul de date gestionează informații despre o aplicație de streaming de muzică și podcast-uri, mai exact despre muzică, artiști, podcast-uri, și cel mai important, utilizatori și activitatea acestora. Utilitatea modelului este dată de posibilitatea de a folosi datele în scopul sugestiilor personalizate pentru fiecare utilizator, astfel îmbunătățind experiența acestora. Datele pot fi folosite și pentru crearea de statistici interne, clasamente publice sau oferirea unei recapitulări anuale pentru fiecare utilizator.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

Sunt respectate anumite restricții și reguli:

- Un episod de podcast poate apartine unui singur podcast
- Un album poate fi lansat de cel puțin un artist.
- Un episod de podcast poate apartine unui singur podcast.
- O melodie poate fi cântata de cel puțin un artist.
- Un playlist poate fi deținut de un singur utilizator.
- O melodie sau un episod de podcast pot fi ascultate de mai multe ori de un utilizator.
- O melodie poate apartine cel mult unui album.
- Un utilizator nu poate avea de mai multe ori aceeași melodie în același playlist.
- Toate atributele numerice trebuie să fie numere pozitive.
- Dacă o melodie face parte dintr-un album, data ei de lansare va fi NULL deoarece aceasta coincide cu data de lansare a albumului.

3. Descrierea entităților, incluzând precizarea cheii primare

Pentru modelul de date ales, entitățile sunt UTILIZATOR, PLAYLIST, MELODIE, ARTIST, ALBUM, PODCAST și EPISOD_PODCAST. Voi prezenta entitățile, precizând cheia primară pentru fiecare dintre acestea.

UTILIZATOR = persoană care utilizează aplicația prin intermediul unui cont. Atributul id_utilizator reprezintă cheia primară a entității.

PLAYLIST = listă de melodii creată de un utilizator cu scopul de a ușura ascultarea melodiilor. Cheia primară a entității este id_playlist.

MELODIE = compoziție muzicală cântată de un artist. Cheia primară a acestei entități este id_melodie.

ARTIST = persoană care încarcă melodii și albume în aplicație pentru a fi ascultate. Această entitate are drept cheie primară id_artist.

ALBUM = colecție de melodii încărcată de un artist. Cheia primară a entității este id_album.

PODCAST = emisiune care de obicei se desfășoară în format audio pentru a pune accentul pe informațiile prezentate, este formată din mai multe episoade. Această entitate are cheia primară id_podcast.

EPISOD_PODCAST = o diviziune a unui podcast. Cheia primară a entității este id_episodes.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Voi realiza prezentarea și descrierea relațiilor pentru modelul de date ales. Acestea au denumiri sugestive pentru a exprima conținutul lor și entitățile pe care le leagă.

EPISOD PODCAST aparține PODCAST = relație dintre EPISOD PODCAST și PODCAST, reflectând legătura dintre acestea (diviziunea unui podcast în episoade). Ea are cardinalitate minimă 0:1 și cardinalitate maximă n:1.

MELODIE aparține ALBUM = relație dintre entitățile MELODIE și ALBUM (din ce album face parte o melodie). Relația are cardinalitate minimă 1:0 și cardinalitate maximă n:1.

UTILIZATOR ascultă EPISOD PODCAST = relație de tip many-to-many dintre entitățile UTILIZATOR și EPISOD PODCAST (istoricul de ascultate al episoadelor de podcast de către toți utilizatorii). Relația are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

UTILIZATOR ascultă MELODIE = relație de tip many-to-many dintre entitățile UTILIZATOR și MELODIE (istoricul de ascultate al melodiei de către toți utilizatorii). Relația are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

UTILIZATOR apreciază PLAYLIST = relație de tip many-to-many dintre entitățile UTILIZATOR și PLAYLIST, reflectând legătura dintre acestea (aprecierea unui playlist face accesarea sa de către un utilizator mai ușoară). Relația are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

UTILIZATOR apreciază ALBUM = relație de tip many-to-many dintre entitățile UTILIZATOR și ALBUM, reflectând legătura dintre acestea (aprecierea unui album face accesarea sa de către un utilizator mai ușoară). Relația are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

UTILIZATOR urmărește ARTIST = relație de tip many-to-many dintre entitățile UTILIZATOR și ARTIST, reflectând legătura dintre acestea (urmărirea unui artist face găsirea acestuia de către un utilizator mai ușoară. De asemenea se poate implementa o funcționalitate în care utilizatorii primesc notificări în momentul în care artiștii urmăriți lansează noi albume/melodii). Relația are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

ARTIST_lansează_ALBUM = relație de tip many-to-many dintre entitățile ARTIST și ALBUM (exprimă posesia unuia sau a mai multor artiști asupra unui album). Relația are cardinalitate minimă 1:0 și cardinalitate maximă m:n.

ARTIST_cântă_MELODIE = relație de tip many-to-many dintre entitățile ARTIST și MELODIE (exprimă posesia unuia sau a mai multor artiști asupra unei melodii). Relația are cardinalitate minimă 1:0 și cardinalitate maximă m:n.

MELODIE_aparține_la_PLAYLIST_deținut_de_UTILIZATOR = relație de tip 3 ce leagă entitățile MELODIE, PLAYLIST și UTILIZATOR, folosită pentru a verifica apartenența unei melodii la un playlist al unui utilizator dar și pentru a găsi toate playlist-urile în care se află o anumită melodie.

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicate, valori posibile ale atributelor.

Entitatea *UTILIZATOR* are ca attribute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator.

nume = variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă numele pe care alții utilizatori îl văd în aplicație.

email = variabilă de tip caracter, de lungime maximă 50, nenulă, unică, care reprezintă adresa de e-mail a utilizatorului.

data_nastere = variabilă de tip dată calendaristică, nenulă, care reprezintă data nașterii utilizatorului respectiv.

tara = variabilă de tip caracter, de lungime maximă 2, nenulă, care reprezintă codul țării unui utilizator, conform Anexei A6, “Codurile țărilor și teritoriilor”.

Entitatea *PLAYLIST* are ca attribute:

id_playlist = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui playlist.

nume = variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă numele unui playlist.

id_creator = variabilă de tip întreg, de lungime maximă 9, nenulă, care reprezintă id-ul creatorului. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

Entitatea *MELODIE* are ca attribute:

id_melodie = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unei melodii.

titlu= variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă titlul unei melodii.

durata = variabilă de tip întreg, de lungime maximă 3, nenulă, care reprezintă durata unei melodii în secunde.

`id_album` = variabilă de tip întreg, de lungime maximă 9, reprezintă id-ul albumului căruia aparține melodia. Acesta poate fi NULL deoarece melodia poate fi un single. Dacă este nenulă, aceasta trebuie să corespundă unei chei primare din tabelul **ALBUM**.

`gen` = variabilă de tip caracter, de lungime maximă 15, care reprezintă genul muzical al unei melodii. Aceasta poate lua valori precum pop, rock, country, funk, hip hop.

`data_lansare` = variabilă de tip dată calendaristică, care reprezintă data de lansare a melodiei. Dacă aceasta face parte dintr-un album, `data_lansare` va fi NULL deoarece această informație poate fi preluată din entitatea **ALBUM**.

Entitatea **ARTIST** are ca attribute:

`id_artist` = variabilă de tip întreg, de lungime maximă 9, reprezintă id-ul unui artist.

`nume` = variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă numele de scenă al artistului / trupei.

`tara` = variabilă de tip caracter, de lungime maximă 2, nenulă, care reprezintă codul țării unui artist, conform Anexei A6, "Codurile țărilor și teritoriilor".

Entitatea **ALBUM** are ca attribute:

`id_album` = variabilă de tip întreg, de lungime maximă 9, reprezintă id-ul unui album.

`titlu` = variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă titlul albumului.

`data_lansare` = variabilă de tip dată calendaristică, nenulă, care reprezintă data de lansare a albumului.

`casa_discuri` = variabilă de tip caracter, de lungime maximă 50, care reprezintă casa de discuri de care aparținea artistul în momentul lansării albumului

Entitatea *PODCAST* are ca atribute:

id_podcast = variabilă de tip întreg, de lungime maximă 9, reprezintă id-ul unui podcast.

nume = variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă numele podcast-ului.

nume_gazda = variabilă de tip caracter, de lungime maximă 50, nenulă, care reprezintă numele gazdei podcast-ului.

limbă = variabilă de tip caracter, de lungime maximă 3, nenulă, care reprezintă limba podcastului. Codificarea se face în conformitate cu ISO 639-2/B.

Entitatea *EPISOD_PODCAST* are ca atribute:

id_episod = variabilă de tip întreg, de lungime maximă 9, reprezintă id-ul unui episod de podcast.

id_podcast = variabilă de tip întreg, nenulă, de lungime maximă 9, nenulă, reprezintă id-ul podcast-ului căruia aparține episodul.

titlu = variabilă de tip caracter, de lungime maximă 75, nenulă, care reprezintă titlul episodului.

durata = variabilă de tip întreg, de lungime maximă 5, nenulă, care reprezintă durata unui episod în secunde.

data_aparitie = variabilă de tip dată calendaristică, nenulă, care reprezintă data de apariție a episodului

Relația UTILIZATOR_ascultă_EPISOD_PODCAST are ca atribute:

id_ascultare_podcast = variabilă de tip întreg, de lungime maximă 9, nenulă, reprezintă id-ul unei ascultări a unui episod de podcast de către un utilizator

id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

id_episod = variabilă de tip întreg, de lungime maximă 9, reprezintă id-ul unui episod de podcast. Atributul trebuie să corespundă unei chei primare din tabelul EPISOD_PODCAST.

data_ascultare = variabilă de tip dată calendaristică, nenulă, care reprezintă data de ascultare a episodului de către utilizator

Relația UTILIZATOR_ascultă_MELODIE are ca atrbute:

id_ascultare_melodie = variabilă de tip întreg, de lungime maximă 9, nenulă, reprezintă id-ul unei ascultări a unei melodii de către un utilizator

id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

id_melodie = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unei melodii. Atributul trebuie să corespundă unei chei primare din tabelul MELODIE.

data_ascultare = variabilă de tip dată calendaristică, nenulă, care reprezintă data de ascultare a melodiei de către utilizator.

Relația UTILIZATOR_apreciază_PLAYLIST are ca atrbute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

id_playlist = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui playlist. Atributul trebuie să corespundă unei chei primare din tabelul PLAYLIST.

Relația UTILIZATOR_apreciază_ALBUM are ca atrbute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

id_album = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui album. Atributul trebuie să corespundă unei chei primare din tabelul ALBUM.

Relația UTILIZATOR_urmărește_ARTIST are ca atrbute:

id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

id_artist = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui artist. Atributul trebuie să corespundă unei chei primare din tabelul ARTIST.

Relația ARTIST_lansează_ALBUM are ca atrbute:

id_artist = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui artist. Atributul trebuie să corespundă unei chei primare din tabelul ARTIST.

id_album = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui album. Atributul trebuie să corespundă unei chei primare din tabelul ALBUM.

Relația ARTIST_cântă_MELODIE are ca atrbute:

id_artist = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui artist. Atributul trebuie să corespundă unei chei primare din tabelul ARTIST.

id_melodie = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unei melodii. Atributul trebuie să corespundă unei chei primare din tabelul MELODIE.

Relația MELODIE_aparține_la_PLAYLIST_deținut_de_UTILIZATOR are ca atrbute:

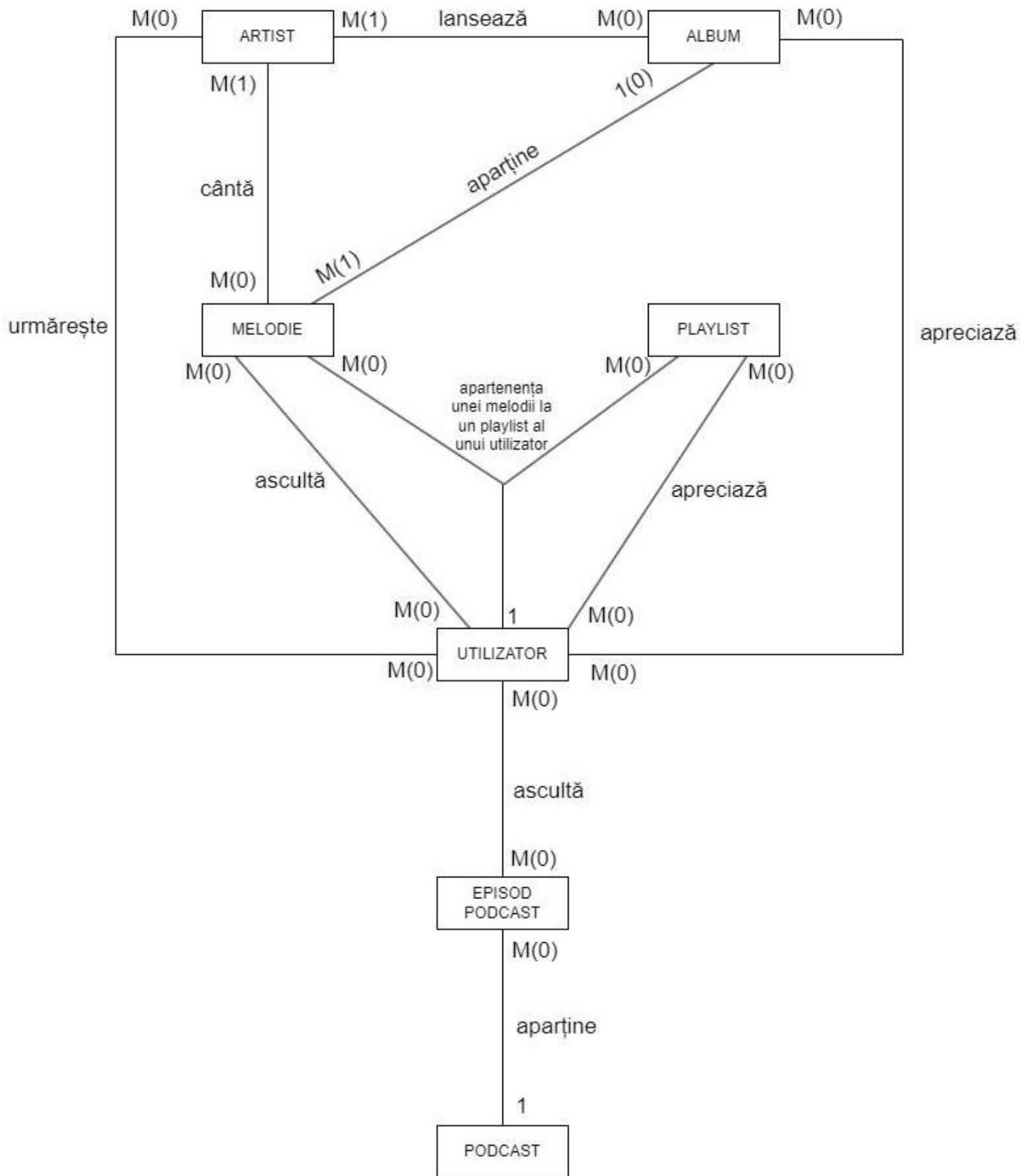
id_melodie = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unei melodii. Atributul trebuie să corespundă unei chei primare din tabelul MELODIE.

id_playlist = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui playlist. Atributul trebuie să corespundă unei chei primare din tabelul PLAYLIST.

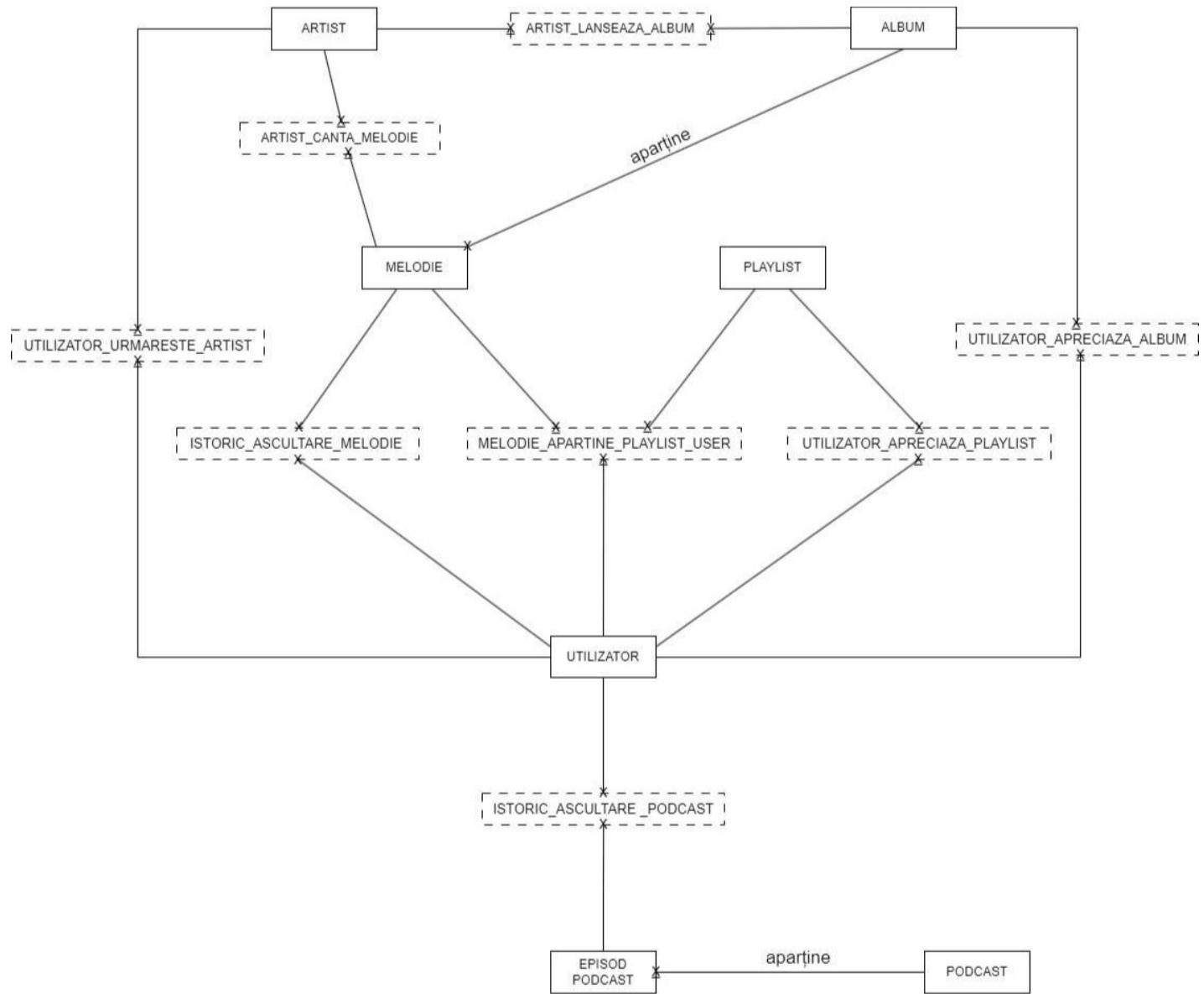
id_utilizator = variabilă de tip întreg, de lungime maximă 9, care reprezintă id-ul unui utilizator. Atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR.

id_data_adaugare = variabilă de tip dată calendaristică, nenulă, care reprezintă data de adăugare a melodiei în playlist de către utilizator.

6. Realizarca diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

UTILIZATOR (id_utilizator(PK), nume, email, data_nastere, tara)

PLAYLIST (id_playlist(PK), id_creator(FK), nume)

MELODIE (id_melodie(PK), id_album(FK), titlu, durata, gen, data_lansare)

ARTIST (id_artist(PK), nume, tara)

ALBUM (id_album(PK), titlu, data_lansare, casa_discuri)

PODCAST (id_podcast(PK), nume, nume_gazda, limba)

EPISOD_PODCAST (id_episod(PK), id_podcast(FK), titlu, durata, data_aparitie)

ISTORIC_ASCULTARE_PODCAST (id_ascultare_podcast(PK), id_utilizator(FK), id_episod(FK), data_ascultare)

ISTORIC_ASCULTARE_MELODIE (id_ascultare_melodie(PK), id_utilizator(FK), id_melodie(FK), data_ascultare)

UTILIZATOR_APRECIAZA_PLAYLIST (id_utilizator(PK, FK), id_playlist(PK, FK))

UTILIZATOR_APRECIAZA_ALBUM (id_utilizator(PK, FK), id_album(PK, FK))

UTILIZATOR_URMARESTE_ARTIST (id_utilizator(PK, FK), id_artist(PK, FK))

ARTIST_LANSEAZA_ALBUM (id_artist(PK, FK), id_album(PK, FK))

ARTIST_CANTA_MELODIE (id_artist(PK, FK), id_melodie(PK, FK))

MELODIE_APARTINE_PLAYLIST_USER (id_melodie(PK, FK), id_playlist(PK, FK), id_utilizator(PK, FK), data_adaugare)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

- Non FN1:

MELODIE				
ID_MELODIE	TITLU	DURATĂ	ARTIȘTI	DATĂ_LANSARE
1	Selecta	206	Alex Velea, Anda Adam	01/01/2005
2	Cai verzi pe pereti	194	Smiley, Alex Velea, Don Baxter	17/01/2017
3	Search & Rescue	272	Drake	11/04/2023

MELODIE				
ID_MELODIE	TITLU	DURATĂ	ARTIST	DATĂ_LANSARE
1	Selecta	206	Alex Velea	01/01/2005
2	Cai verzi pe pereti	194	Smiley	17/01/2017
1	Selecta	206	Anda Adam	01/01/2005
3	Search & Rescue	272	Drake	11/04/2023
1	Cai verzi pe pereti	194	Don Baxter	17/01/2017
1	Cai verzi pe pereti	194	Alex Velea	17/01/2017

FN1 reprezintă obținerea atributelor atomice, adică attribute care nu mai pot fi divizate. În exemplul de mai sus se poate observa cum fiecarei melodii îi corespund unul sau mai mulți artiști. Descompunerea este realizată prin introducerea datelor despre melodie pentru fiecare artist care o cântă. Deși acest lucru adaugă redundanță, acum fiecare atribut este atomic.

În general, pentru a obține FN1, un atribut “nume_persoană” (care reprezintă întregul nume al persoanei) este descompus în “nume_persoană” și “prenume_persoană”. Totuși, în cazul de față, atributul “artist” nu poate fi descompus astfel întrucât reprezintă numele de scenă al artistului (sau al unei trupe).

- Non FN2:

MELODIE					
ID_MELODIE	TITLU	DURATĂ	ARTIST	DATA_LANSARE	ȚARĂ_ARTIST
1	Selecta	206	Alex Velea	01/01/2005	RO
2	Cai verzi pe pereți	194	Smiley	17/01/2017	AT
1	Selecta	206	Anda Adam	01/01/2005	BG
3	Minim Doi	272	Alex Velea	11/04/2023	RO
1	Cai verzi pe pereți	194	Don Baxter	17/01/2017	UK
1	Cai verzi pe pereți	194	Alex Velea	17/01/2017	RO

- FN2:

MELODIE					ARTIST		
ID_MELODIE	TITLU	DURATĂ	ID_ARTIST	DATA_LANSARE	ID_ARTIST	NUME	ȚARĂ
1	Selecta	206	1	01/01/2005	1	Alex Velea	RO
2	Cai verzi pe pereți	194	2	17/01/2017	2	Smiley	AT
1	Selecta	206	4	01/01/2005	3	Don Baxter	UK
3	Minim Doi	272	1	11/04/2023	4	Anda Adam	BG
1	Cai verzi pe pereți	194	3	17/01/2017			
1	Cai verzi pe pereți	194	1	17/01/2017			

Un tabel se află în FN2 dacă se află în FN1 și fiecare atribut care nu participă la cheia primară este dependent de întreaga cheie primară. Pentru a exemplifica acest lucru am adăugat la entitatea MELODIE un atribut numit “țară_artist”. Acest atribut nu depinde de cheia primară a entității MELODIE, respectiv “id_melodie”, o cheie primară artificială. Pentru ca tabela să fie în FN2, adăugăm o nouă entitate, ARTIST, unde atribuim fiecărui artist un id și țara acestuia.

- Non FN3:

MELODIE				
ID_MELODIE	TITLU	DURATĂ	ID_ALBUM	NUME_ALBUM
1	20 Min	220	1	Luv Is Rage 2
2	Neon Guts	194	1	Luv Is Rage 2
3	Scopul	151	2	Ştefan III
4	Minim Doi	272	-	-
5	Slayer	176	3	Slayer
6	Asumă-Ńi	221	2	Ştefan III

- FN3:

MELODIE			
ID_MELODIE	TITLU	DURATĂ	ID_ALBUM
1	20 Min	220	1
2	Neon Guts	194	1
3	Scopul	151	2
4	Minim Doi	272	-
5	Slayer	176	3
6	Asumă-Ńi	221	2

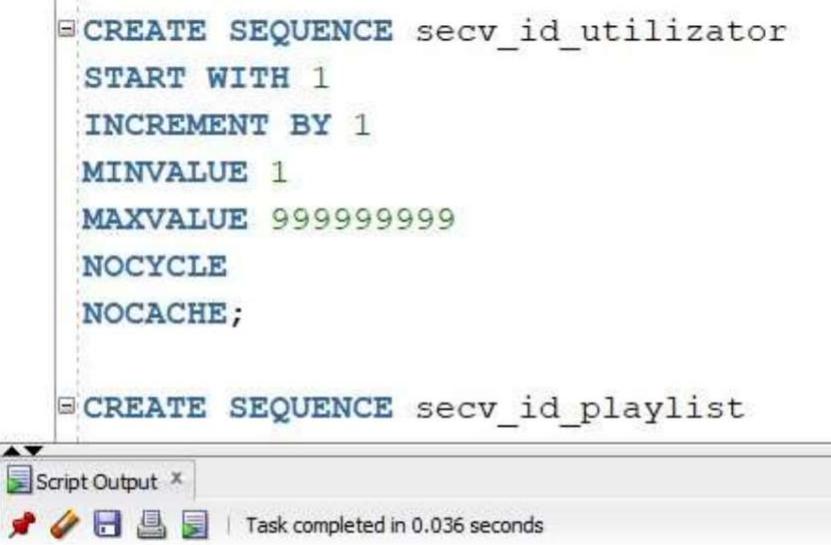
ALBUM	
ID_ALBUM	NUME_ALBUM
1	Luv Is Rage 2
2	Ştefan III
3	Slayer

Un tabel se află în FN3 dacă acesta se află în FN2 și nu există dependențe tranzitive. În exemplul de mai sus, "nume_album" poate fi derivat din "id_album", astfel se creează o dependență tranzitivă: **id_melodie -> id_album -> nume_album**. Pentru a rezolva această situație, se creează tabela ALBUM conform exemplului de mai sus și în tabela MELODIE "id_album" devine cheie externă.

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele.

```
CREATE SEQUENCE secv_id_utilizator
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
NOCYCLE
NOCACHE;

CREATE SEQUENCE secv_id_playlist
```



The screenshot shows the Oracle SQL Developer interface. At the top, there are two code snippets for creating sequences. Below them is a 'Script Output' window with a toolbar. The window displays the message 'Task completed in 0.036 seconds'. The status bar at the bottom indicates 'Sequence SECV_ID_UTILIZATOR created.'

Sequence SECV_ID_UTILIZATOR created.

```
CREATE SEQUENCE secv_id_utilizator
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
NOCYCLE
NOCACHE;
```

```
CREATE SEQUENCE secv_id_playlist
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
```

NOCYCLE

NOCACHE;

CREATE SEQUENCE secv_id_melodie

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 999999999

NOCYCLE

NOCACHE;

CREATE SEQUENCE secv_id_artist

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 999999999

NOCYCLE

NOCACHE;

CREATE SEQUENCE secv_id_album

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 999999999

NOCYCLE

NOCACHE;

```
CREATE SEQUENCE secv_id_podcast
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
NOCYCLE
NOCACHE;
```

```
CREATE SEQUENCE secv_id_ep_podcast
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
NOCYCLE
NOCACHE;
```

```
CREATE SEQUENCE secv_id_asc_podcast
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
NOCYCLE
NOCACHE;
```

```
CREATE SEQUENCE secv_id_asc_melodie
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 999999999
NOCYCLE
NOCACHE;
```

**11. Crearea tabelelor în SQL și inserarea de date coerente
în fiecare dintre acestea (minimum 5 înregistrări în fiecare
tabel neasociativ și minimum 10 înregistrări în tabelele
asociative).**

ID_UTILIZATOR	NUME	EMAIL	DATA_NASTERE	TARA
1	1 ingeaua	ingeaua.alexandru@gmail.com	15-MAY-2003 00:00:00	RO
2	2 Ionut45	ionutfotbal6@yahoo.com	05-JUL-2006 00:00:00	RO
3	3 Marius	marius.boieriu@gmail.com	29-DEC-1997 00:00:00	RO
4	4 JohnnyTheBoy	johndoe@gmail.com	20-MAY-2001 00:00:00	UK
5	5 NelkaMM3	nelka.miroslava@gmail.com	06-JUN-1976 00:00:00	PL

ID_PLAYLIST	ID_CREA...	NUME
1	1	1 osky
2	2	1 musique
3	3	2 RandBBBB
4	4	4 party songs
5	5	5 LUV is luv

ID_ARTIST	NUME	TARA
1	1 Lil Uzi Vert	US
2	2 Ian	RO
3	3 OSCAR	RO
4	4 Calinacho	RO
5	5 Pharrell Williams	US
6	6 Amuly	RO
7	7 Azteca	RO
8	8 Future	US
9	9 PRNY	RO

ID_ALBUM	TITLU	DATA_LANSARE	CASA_DISCURI
1	1 Apa	29-OCT-2020 00:00:00	OSKY
2	2 Luv Is Rage 2	25-AUG-2017 00:00:00	Atlantic Recording Corporation
3	3 intr-o buna zi o sa fim bine	25-NOV-2022 00:00:00	nobody:nothing
4	4 Slayer	21-JUN-2019 00:00:00	Ocult Records
5	5 Sin City	21-DEC-2020 00:00:00	Seek Music
6	6 616	08-APR-2022 00:00:00	(null)
7	7 Pluto x Baby Pluto	13-NOV-2020 00:00:00	Atlantic Recording Corporation
8	8 GEISHA	23-NOV-2020 00:00:00	Seek Music

ID_MELODIE	TITLU	DURATA	ID_ALBUM	GEN	DATA_LANSARE
1	1 Tunechi	242	1 Trap	(null)	
2	2 Lacrimi	224	1 Trap	(null)	
3	3 Aripi	180	1 Trap	(null)	
4	4 444+222	247	2 Emo...	(null)	
5	5 Neon Guts	258	2 Emo...	(null)	
6	6 wattba	221	3 R&B	(null)	
7	7 rana	195	3 R&B	(null)	
8	8 Biznis	134	4 Trap	(null)	
9	9 Hell-O	194	4 Trap	(null)	
10	10 Ne Schimbam	163	5 Trap	(null)	
11	11 Vampiri	199	5 Trap	(null)	
12	12 Teanc	189	5 Rap	(null)	
13	13 A-U-Freestyle	135	6 Trap	(null)	
14	14 UFO	135	6 Trap	(null)	
15	15 Lullaby	261	7 Emo...	(null)	
16	16 Patek	280	7 Trap	(null)	
17	17 Plastic	193	7 Rap	(null)	
18	18 Oui, oui	155	8 Trap	(null)	
19	19 Rocco	107	8 Trap	(null)	
20	20 Orasul Pacatelor	214	8 Trap	(null)	
21	21 Nefiu	185	(null) Trap	19-DEC-2018 00:00:00	
22	22 Double Up	173	(null) Trap	08-NOV-2020 00:00:00	
23	23 Never Recover	116	(null) Trap	31-MAR-2019 00:00:00	
24	24 Roceri	166	(null) R&B	30-JUL-2021 00:00:00	

ID_PODCAST	NUME	NUME_GAZDA	LIMBA
1	1 The Joe Rogan Experience	Joe Rogan	ENG
2	2 Mind Architect	Mind Architect	RUM
3	3 Fain & Simplus Podcast	Mihai Morar	RUM
4	4 Huberman Lab	Scicomm Media	ENG
5	5 Psihologi la Cafea	Cristina Ursu	RUM

	ID_ASCULTARE_PODCAST	ID_UTILIZATOR	ID_EPISOD	DATA_ASCULTARE
1	1	1	3	08-MAY-2023 19:43:21
2	2	2	3	08-MAY-2023 19:43:21
3	3	3	3	08-MAY-2023 19:43:21
4	4	4	3	08-MAY-2023 19:43:21
5	5	2	7	08-MAY-2023 19:43:21
6	6	3	7	08-MAY-2023 19:43:21
7	7	3	9	08-MAY-2023 19:43:21
8	8	3	8	08-MAY-2023 19:43:21
9	9	3	10	08-MAY-2023 19:43:21
10	10	2	3	08-MAY-2023 19:43:21
11	11	4	3	08-MAY-2023 19:43:21
12	12	2	7	08-MAY-2023 19:43:21
13	13	3	1	08-MAY-2023 19:43:21
14	14	2	1	08-MAY-2023 19:43:21
15	15	3	2	08-MAY-2023 19:43:21
16	16	2	2	08-MAY-2023 19:43:21

	ID_ASCULTARE_MELODIE	ID_UTILIZATOR	ID_MELODIE	DATA_ASCULTARE
1	1	1	1	08-MAY-2023 19:43:21
2	2	2	6	08-MAY-2023 19:43:21
3	3	3	6	08-MAY-2023 19:43:21
4	4	4	6	08-MAY-2023 19:43:21
5	5	5	6	08-MAY-2023 19:43:21
6	6	4	9	08-MAY-2023 19:43:21
7	7	3	7	08-MAY-2023 19:43:21
8	8	5	11	08-MAY-2023 19:43:21
9	9	4	11	08-MAY-2023 19:43:21
10	10	3	2	08-MAY-2023 19:43:21
11	11	2	3	08-MAY-2023 19:43:21
12	12	1	4	08-MAY-2023 19:43:21
13	13	2	2	08-MAY-2023 19:43:21
14	14	1	1	08-MAY-2023 19:43:21
15	15	1	4	12-JAN-2022 14:45:27
16	16	1	5	15-FEB-2022 18:35:27
17	17	1	4	05-MAR-2022 12:32:27
18	18	1	4	01-APR-2022 14:45:27
19	19	1	4	01-MAY-2022 16:25:27
20	20	2	5	25-JUN-2022 18:35:27
21	21	2	5	18-JUL-2022 18:35:27
22	22	2	4	04-APR-2022 18:35:27
23	23	1	4	20-MAY-2022 18:35:00
24	24	3	4	15-JUN-2022 18:35:27
25	25	4	8	15-JUL-2022 20:35:40
26	26	3	8	16-SEP-2022 10:53:25
27	27	3	8	19-OCT-2022 12:35:27
28	28	3	8	12-NOV-2022 18:20:30
29	29	4	4	10-DEC-2022 20:30:20

ID_ARTIST	ID_ALBUM
1	3
2	1
3	4
4	2
5	3
6	6
7	7
8	1
9	8
10	9

ID_UTILIZATOR	ID_PLAYLIST
1	3
2	4
3	5
4	1
5	2
6	3
7	1
8	1
9	2
10	3

ID_EPISOD	ID_PODCAST	TITLU	DURATA	DATA_APARITIE
1	1	1 #1976 - James Fox	8917	26-APR-2023 00:00:00
2	2	1 #1889 - Dr. Phil	1226	728-OCT-2022 00:00:00
3	3	2 Introducere in Arhitectura Cerebrală	1241	01-NOV-2019 00:00:00
4	4	2 Cum sa imbatranesti frumos	1608	05-JUN-2020 00:00:00
5	5	3 CUM SA FACI BANI IN CRIZA	6830	08-APR-2022 00:00:00
6	6	3 DECLARATIE PENTRU VIATA. GEORGE BALTA	4123	29-JUL-2021 00:00:00
7	7	4 How to stop Headaches	8760	06-FEB-2023 00:00:00
8	8	4 Understand and improve memory	7800	16-MAY-2022 00:00:00
9	9	5 Anxietate	1295	20-MAR-2023 00:00:00
10	10	5 Hipnoza	1526	22-JUN-2022 00:00:00

ID_UTILIZATOR	ID_ARTIST
1	1
2	1
3	1
4	1
5	4
6	4
7	4
8	5
9	5
10	5

ID_UTILIZATOR	ID_ALBUM
1	1
2	4
3	3
4	8
5	2
6	3
7	5
8	3
9	3
10	1

ID_ARTIST	ID_MELODIE
1	2
2	3
3	2
4	3
5	6
6	1
7	1
8	5
9	4
10	4
11	2
12	2
13	3
14	6
15	3
16	6
17	3
18	6
19	7
20	7
21	1
22	1
23	8
24	1
25	8
26	9
27	7
28	3
29	9
30	6
31	2
32	9
33	6
34	3
35	2
36	7

	ID_MELCDIE	ID_PLAYLIST	ID_UTILIZATOR	DATA ADAUGARE
1	1	1	1	1 08-MAY-2023 19:43:21
2	2	1	1	1 08-MAY-2023 19:43:21
3	3	1	1	1 08-MAY-2023 19:43:21
4	10	1	1	1 08-MAY-2023 19:43:21
5	11	1	1	1 08-MAY-2023 19:43:21
6	12	1	1	1 08-MAY-2023 19:43:21
7	16	2	1	1 08-MAY-2023 19:43:21
8	21	2	1	1 08-MAY-2023 19:43:21
9	24	2	1	1 08-MAY-2023 19:43:21
10	6	3	1	2 08-MAY-2023 19:43:21
11	7	3	1	2 08-MAY-2023 19:43:21
12	24	3	1	2 08-MAY-2023 19:43:21
13	2	3	1	2 08-MAY-2023 19:43:21
14	21	4	1	4 08-MAY-2023 19:43:21
15	22	4	1	4 08-MAY-2023 19:43:21
16	23	4	1	4 08-MAY-2023 19:43:21
17	8	4	1	4 08-MAY-2023 19:43:21
18	9	4	1	4 08-MAY-2023 19:43:21
19	13	4	1	4 08-MAY-2023 19:43:21
20	4	5	1	5 08-MAY-2023 19:43:21
21	5	5	1	5 08-MAY-2023 19:43:21
22	15	5	1	5 08-MAY-2023 19:43:21
23	16	5	1	5 08-MAY-2023 19:43:21
24	17	5	1	5 08-MAY-2023 19:43:21

Screenshot-urile precedente au fost făcute după rularea cererii:

SELECT * FROM -tabel dorit-

```
CREATE TABLE utilizator(
    id_utilizator NUMBER(9) CONSTRAINT pk_id_user PRIMARY KEY,
    nume VARCHAR2(50) CONSTRAINT null_nume_user NOT NULL,
    email VARCHAR2(50) CONSTRAINT null_email_user NOT NULL,
    data_nastere DATE CONSTRAINT null_data_user NOT NULL,
    tara VARCHAR2(2) CONSTRAINT null_tara_user NOT NULL,
    CONSTRAINT unq_email_user UNIQUE(email));
```

```
CREATE TABLE playlist(
    id_playlist NUMBER(9) CONSTRAINT pk_id_plist PRIMARY KEY,
    id_creator NUMBER(9) CONSTRAINT null_creator_plist NOT
NULL,
    nume VARCHAR2(50) CONSTRAINT null_nume_plist NOT NULL,
    CONSTRAINT fk_playlist_creator FOREIGN KEY(id_creator)
REFERENCES utilizator(id_utilizator) ON DELETE CASCADE);
```

```
CREATE TABLE album(
    id_album NUMBER(9) CONSTRAINT pk_id_album PRIMARY KEY,
    titlu VARCHAR2(50) CONSTRAINT null_nume_album NOT NULL,
    data_lansare DATE CONSTRAINT null_data_album NOT NULL,
    casa_discuri VARCHAR2(50));
```

```
CREATE TABLE melodie(
    id_melodie NUMBER(9) CONSTRAINT pk_id_song PRIMARY KEY,
    titlu VARCHAR2(50) CONSTRAINT null_titlu_song NOT NULL,
    durata NUMBER(3) CONSTRAINT null_durata_song NOT NULL,
    id_album NUMBER(9) CONSTRAINT fk_song_album REFERENCES
    album(id_album) ON DELETE CASCADE,
    gen VARCHAR2(15),
    data_lansare date,
    constraint song_lansare CHECK (data_lansare IS NOT NULL AND
    id_album IS NULL
    OR id_album IS NOT NULL AND data_lansare IS NULL));
```

```
CREATE TABLE artist(
    id_artist NUMBER(9) CONSTRAINT pk_id_artist PRIMARY KEY,
    nume VARCHAR2(50) CONSTRAINT null_nume_artist NOT NULL,
    tara VARCHAR2(2) CONSTRAINT null_tara_artist NOT NULL);
```

```
CREATE TABLE podcast(
    id_podcast NUMBER(9) CONSTRAINT pk_id_podcast PRIMARY
    KEY,
    nume VARCHAR2(50) CONSTRAINT null_nume_podcast NOT
    NULL,
    nume_gazda VARCHAR2(50) CONSTRAINT null_gazda_podcast
    NOT NULL,
    limba VARCHAR2(3) CONSTRAINT null_limba_podcast NOT
    NULL);
```

```
CREATE TABLE episod_podcast(
    id_episod NUMBER(9) CONSTRAINT pk_id_episod PRIMARY KEY,
    id_podcast NUMBER(9) CONSTRAINT null_id_podcast NOT NULL,
    titlu VARCHAR2(50) CONSTRAINT null_titlu_ep NOT NULL,
    durata NUMBER(5) CONSTRAINT null_durata_ep NOT NULL,
    data_aparitie DATE CONSTRAINT null_data_ep NOT NULL,
    CONSTRAINT fk_episod_podcast FOREIGN KEY (id_podcast)
    REFERENCES podcast(id_podcast) ON DELETE CASCADE);
```

```
CREATE TABLE istoric_ascultare_podcast (
    id_ascultare_podcast NUMBER(9) CONSTRAINT pk_id_asc_pod
    PRIMARY KEY,
    id_utilizator NUMBER(9),
    id_episod NUMBER(9) CONSTRAINT null_id_asc_pod_podcast
    NOT NULL,
    data_ascultare DATE CONSTRAINT null_data_asc_pod NOT NULL,
    CONSTRAINT fk_asc_pod_id_user FOREIGN KEY (id_utilizator)
    REFERENCES utilizator(id_utilizator) ON DELETE SET NULL,
    CONSTRAINT fk_asc_pod_id_ep FOREIGN KEY (id_episod)
    REFERENCES episod_podcast(id_episod) ON DELETE CASCADE);
```

```
CREATE TABLE istoric_ascultare_melodie (
    id_ascultare_melodie NUMBER(9) CONSTRAINT pk_id_asc_song
    PRIMARY KEY,
    id_utilizator NUMBER(9),
    id_melodie NUMBER(9) CONSTRAINT null_id_asc_song_song NOT
    NULL,
    data_ascultare DATE CONSTRAINT null_data_asc_song NOT NULL,
```

```
    CONSTRAINT fk_asc_song_id_user FOREIGN KEY (id_utilizator)
REFERENCES utilizator(id_utilizator) ON DELETE SET NULL,
    CONSTRAINT fk_asc_song_id_song FOREIGN KEY (id_melodie)
REFERENCES melodie(id_melodie) ON DELETE CASCADE);
```

```
CREATE TABLE utilizator_apreciaza_playlist (
    id_utilizator NUMBER(9) CONSTRAINT fk_userplist_user
REFERENCES utilizator(id_utilizator) ON DELETE CASCADE,
    id_playlist NUMBER(9) CONSTRAINT fk_userplist_plist
REFERENCES playlist(id_playlist) ON DELETE CASCADE,
    CONSTRAINT pk_userplist PRIMARY KEY (id_utilizator,
id_playlist));
```

```
CREATE TABLE utilizator_apreciaza_album (
    id_utilizator NUMBER(9) CONSTRAINT fk_useralbum_user
REFERENCES utilizator(id_utilizator) ON DELETE CASCADE,
    id_album NUMBER(9) CONSTRAINT fk_useralbum_album
REFERENCES album(id_album) ON DELETE CASCADE,
    CONSTRAINT pk_useralbum PRIMARY KEY (id_utilizator,
id_album));
```

```
CREATE TABLE utilizator_urmareste_artist (
    id_utilizator NUMBER(9) CONSTRAINT fk_userartist_user
REFERENCES utilizator(id_utilizator) ON DELETE CASCADE,
    id_artist NUMBER(9) CONSTRAINT fk_userartist_artist
REFERENCES artist(id_artist) ON DELETE CASCADE,
    CONSTRAINT pk_userartist PRIMARY KEY (id_utilizator,
id_artist));
```

```

CREATE TABLE artist_lanseaza_album (
    id_artist NUMBER(9) CONSTRAINT fk_artistalbum_artist
REFERENCES artist(id_artist) ON DELETE CASCADE,
    id_album NUMBER(9) CONSTRAINT fk_artistalbum_album
REFERENCES album(id_album) ON DELETE CASCADE,
    CONSTRAINT pk_artistalbum PRIMARY KEY (id_artist, id_album));

```

```

CREATE TABLE artist_canta_melodie (
    id_artist NUMBER(9) CONSTRAINT fk_artistmelodie_artist
REFERENCES artist(id_artist) ON DELETE CASCADE,
    id_melodie NUMBER(9) CONSTRAINT fk_artistmelodie_song
REFERENCES melodie(id_melodie) ON DELETE CASCADE,
    CONSTRAINT pk_artistmelodie PRIMARY KEY (id_artist,
    id_melodie));

```

```

CREATE TABLE melodie_apartine_playlist_user (
    id_melodie NUMBER(9) CONSTRAINT fk_rel3_song REFERENCES
melodie(id_melodie) ON DELETE CASCADE,
    id_playlist NUMBER(9) CONSTRAINT fk_rel3_plist REFERENCES
playlist(id_playlist) ON DELETE CASCADE,
    id_utilizator NUMBER(9) CONSTRAINT fk_rel3_user REFERENCES
utilizator(id_utilizator) ON DELETE CASCADE,
    data_adaugare DATE CONSTRAINT null_rel3_data NOT NULL,
    CONSTRAINT pk_rel3 PRIMARY KEY (id_melodie, id_playlist,
    id_utilizator));

```

```
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Lil Uzi Vert',  
'US');  
  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Ian', 'RO');  
  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'OSCAR',  
'RO');  
  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Calinacho',  
'RO');  
  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Pharrell  
Williams', 'US');  
  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Amuly', 'RO');  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Azteca', 'RO');  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'Future', 'US');  
INSERT INTO artist VALUES (secv_id_artist.NEXTVAL, 'PRNY', 'RO');  
  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'Apa',  
TO_DATE('29-october-2020'), 'OSKY');  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'Luv Is Rage  
2', TO_DATE('25-august-2017'), 'Atlantic Recording Corporation');  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'intr-o buna  
zi o sa fim bine', TO_DATE('25-november-2021'), 'nobody:nothing');  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'Slayer',  
TO_DATE('21-june-2019'), 'Ocult Records');  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'Sin City',  
TO_DATE('21-december-2020'), 'Seek Music');  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, '616',  
TO_DATE('8-april-2021'), NULL);  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'Pluto x  
Baby Pluto', TO_DATE('13-november-2020'), 'Atlantic Recording  
Corporation');  
  
INSERT INTO album VALUES (secv_id_album.NEXTVAL, 'GEISHA',  
TO_DATE('23-november-2020'), 'Seek Music');
```

```
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL,
'Tunechi', 242, 1, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Lacrimi',
224, 1, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Aripi',
180, 1, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL,
'444+222', 247, 2, 'Emo Rap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Neon
Guts', 258, 2, 'Emo Rap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'wattba',
221, 3, CONCAT('R', CONCAT(CHR(38), 'B')), NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'rana',
195, 3, CONCAT('R', CONCAT(CHR(38), 'B')), NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Biznis',
134, 4, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Hell-O',
194, 4, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Ne
Schimbam', 163, 5, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL,
'Vampiri', 199, 5, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL,
'Teanc', 189, 5, 'Rap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL,
'A-U-Freestyle', 135, 6, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'UFO',
135, 6, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Lullaby',
261, 7, 'Emo Rap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Patek',
280, 7, 'Trap', NULL);

INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Plastic',
```

```

193, 7, 'Rap', NULL);
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Oui,
oui', 155, 8, 'Trap', NULL);
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Rocco',
107, 8, 'Trap', NULL);
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Orasul
Pacatelor', 214, 8, 'Trap', NULL);
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Nefiu',
185, NULL, 'Trap', TO_DATE('19-december-2018'));
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Double
Up', 173, NULL, 'Trap', TO_DATE('8-november-2020'));
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Never
Recover', 116, NULL, 'Trap', TO_DATE('31-march-2019'));
INSERT INTO melodie VALUES (secv_id_melodie.NEXTVAL, 'Roceri',
166, NULL, CONCAT('R', CONCAT(CHR(38), 'B')),
TO_DATE('30-july-2021'));

INSERT INTO utilizator VALUES (secv_id_utilizator.NEXTVAL,
'ingeaua', 'ingeaua.alexandru@gmail.com', TO_DATE('15-may-2003'),
'RO');
INSERT INTO utilizator VALUES (secv_id_utilizator.NEXTVAL,
'Ionut45Fotbal3', 'ionutfotbal6@yahoo.com', TO_DATE('5-july-2006'),
'RO');
INSERT INTO utilizator VALUES (secv_id_utilizator.NEXTVAL,
'MariuS', 'marius.boieriu@gmail.com', TO_DATE('29-december-1997'),
'RO');
INSERT INTO utilizator VALUES (secv_id_utilizator.NEXTVAL,
'JohnnyTheBoy', 'johndoe@gmail.com', TO_DATE('20-may-2001'),
'UK');
INSERT INTO utilizator VALUES (secv_id_utilizator.NEXTVAL,
'NelkaMM3', 'nelka.miroslava@gmail.com', TO_DATE('6-june-1976'),
'PL');

```

INSERT INTO podcast VALUES (secv_id_podcast.NEXTVAL, 'The Joe Rogan Experience', 'Joe Rogan', 'ENG');

INSERT INTO podcast VALUES (secv_id_podcast.NEXTVAL, 'Mind Architect', 'Mind Architect', 'RUM');

INSERT INTO podcast VALUES (secv_id_podcast.NEXTVAL, CONCAT('Fain ', CONCAT(CHR(38), 'Simplu Podcast')), 'Mihai Morar', 'RUM');

INSERT INTO podcast VALUES (secv_id_podcast.NEXTVAL, 'Huberman Lab', 'Scicomm Media', 'ENG');

INSERT INTO podcast VALUES (secv_id_podcast.NEXTVAL, 'Psiholog la Cafea', 'Cristina Ursu', 'RUM');

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 1, '#1976 - James Fox', 8917,
TO_DATE('26-april-2023'));

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 1, '#1889 - Dr. Phil', 12267,
TO_DATE('28-october-2022'));

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 2, 'Introducere in Arhitectura
Cerebrala', 1241, TO_DATE('1-november-2019'));

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 2, 'Cum sa imbatranesti frumos', 1608,
TO_DATE('5-june-2020'));

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 3, 'CUM SA FACI BANI IN CRIZA',
6830, TO_DATE('8-april-2022'));

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 3, 'DECLARATIE PENTRU VIATA.
GEORGE BALTA', 4123, TO_DATE('29-july-2021'));

INSERT INTO episod_podcast VALUES
(secv_id_ep_podcast.NEXTVAL, 4, 'How to stop Headaches', 8760,
TO_DATE('6-february-2023'));

```
INSERT INTO episod_podcast VALUES  
(secv_id_ep_podcast.NEXTVAL, 4, 'Understand and improve memory',  
7800, TO_DATE('16-may-2022'));
```

```
INSERT INTO episod_podcast VALUES  
(secv_id_ep_podcast.NEXTVAL, 5, 'Anxietate', 1295,  
TO_DATE('20-march-2023'));
```

```
INSERT INTO episod_podcast VALUES  
(secv_id_ep_podcast.NEXTVAL, 5, 'Hipnoza', 1526,  
TO_DATE('22-june-2022'));
```

```
INSERT INTO playlist VALUES (secv_id_playlist.NEXTVAL, 1, 'osky');
```

```
INSERT INTO playlist VALUES (secv_id_playlist.NEXTVAL, 1,  
'musique');
```

```
INSERT INTO playlist VALUES (secv_id_playlist.NEXTVAL, 2,  
'RandBBBB');
```

```
INSERT INTO playlist VALUES (secv_id_playlist.NEXTVAL, 4, 'party  
songs');
```

```
INSERT INTO playlist VALUES (secv_id_playlist.NEXTVAL, 5, 'LUV is  
luv');
```

```
INSERT INTO artist_lanseaza_album VALUES(3, 1);
```

```
INSERT INTO artist_lanseaza_album VALUES(1, 2);
```

```
INSERT INTO artist_lanseaza_album VALUES(4, 3);
```

```
INSERT INTO artist_lanseaza_album VALUES(2, 4);
```

```
INSERT INTO artist_lanseaza_album VALUES(3, 5);
```

```
INSERT INTO artist_lanseaza_album VALUES(6, 5);
```

```
INSERT INTO artist_lanseaza_album VALUES(7, 6);
```

```
INSERT INTO artist_lanseaza_album VALUES(1, 7);
```

```
INSERT INTO artist_lanseaza_album VALUES(8, 7);
```

```
INSERT INTO artist_lanseaza_album VALUES(9, 8);
```

```
INSERT INTO artist_canta_melodie VALUES(2, 1);
INSERT INTO artist_canta_melodie VALUES(3, 1);
INSERT INTO artist_canta_melodie VALUES(3, 2);
INSERT INTO artist_canta_melodie VALUES(3, 3);
INSERT INTO artist_canta_melodie VALUES(6, 3);
INSERT INTO artist_canta_melodie VALUES(1, 4);
INSERT INTO artist_canta_melodie VALUES(1, 5);
INSERT INTO artist_canta_melodie VALUES(5, 5);
INSERT INTO artist_canta_melodie VALUES(4, 6);
INSERT INTO artist_canta_melodie VALUES(4, 7);
INSERT INTO artist_canta_melodie VALUES(2, 8);
INSERT INTO artist_canta_melodie VALUES(2, 9);
INSERT INTO artist_canta_melodie VALUES(3, 10);
INSERT INTO artist_canta_melodie VALUES(6, 10);
INSERT INTO artist_canta_melodie VALUES(3, 11);
INSERT INTO artist_canta_melodie VALUES(6, 11);
INSERT INTO artist_canta_melodie VALUES(3, 12);
INSERT INTO artist_canta_melodie VALUES(6, 12);
INSERT INTO artist_canta_melodie VALUES(7, 13);
INSERT INTO artist_canta_melodie VALUES(7, 14);
INSERT INTO artist_canta_melodie VALUES(1, 15);
INSERT INTO artist_canta_melodie VALUES(1, 16);
INSERT INTO artist_canta_melodie VALUES(8, 16);
INSERT INTO artist_canta_melodie VALUES(1, 17);
INSERT INTO artist_canta_melodie VALUES(8, 17);
INSERT INTO artist_canta_melodie VALUES(9, 18);
INSERT INTO artist_canta_melodie VALUES(7, 18);
```

```
INSERT INTO artist_canta_melodie VALUES(3, 18);
INSERT INTO artist_canta_melodie VALUES(9, 19);
INSERT INTO artist_canta_melodie VALUES(6, 19);
INSERT INTO artist_canta_melodie VALUES(2, 19);
INSERT INTO artist_canta_melodie VALUES(9, 20);
INSERT INTO artist_canta_melodie VALUES(6, 20);
INSERT INTO artist_canta_melodie VALUES(3, 20);
INSERT INTO artist_canta_melodie VALUES(2, 21);
INSERT INTO artist_canta_melodie VALUES(7, 21);
INSERT INTO artist_canta_melodie VALUES(2, 22);
INSERT INTO artist_canta_melodie VALUES(6, 22);
INSERT INTO artist_canta_melodie VALUES(2, 23);
INSERT INTO artist_canta_melodie VALUES(7, 23);
INSERT INTO artist_canta_melodie VALUES(6, 24);
INSERT INTO artist_canta_melodie VALUES(4, 24);
INSERT INTO artist_canta_melodie VALUES(3, 24);
```

```
INSERT INTO utilizator_urmareste_artist VALUES(1, 5);
INSERT INTO utilizator_urmareste_artist VALUES(1, 4);
INSERT INTO utilizator_urmareste_artist VALUES(1, 3);
INSERT INTO utilizator_urmareste_artist VALUES(1, 8);
INSERT INTO utilizator_urmareste_artist VALUES(4, 1);
INSERT INTO utilizator_urmareste_artist VALUES(4, 2);
INSERT INTO utilizator_urmareste_artist VALUES(4, 7);
INSERT INTO utilizator_urmareste_artist VALUES(5, 8);
INSERT INTO utilizator_urmareste_artist VALUES(5, 5);
INSERT INTO utilizator_urmareste_artist VALUES(5, 2);
```

```
INSERT INTO utilizator_apreciaza_album VALUES(1, 1);
INSERT INTO utilizator_apreciaza_album VALUES(1, 4);
INSERT INTO utilizator_apreciaza_album VALUES(1, 3);
INSERT INTO utilizator_apreciaza_album VALUES(1, 8);
INSERT INTO utilizator_apreciaza_album VALUES(2, 8);
INSERT INTO utilizator_apreciaza_album VALUES(2, 3);
INSERT INTO utilizator_apreciaza_album VALUES(2, 5);
INSERT INTO utilizator_apreciaza_album VALUES(3, 3);
INSERT INTO utilizator_apreciaza_album VALUES(4, 3);
INSERT INTO utilizator_apreciaza_album VALUES(4, 1);

INSERT INTO melodie_apartine_playlist_user VALUES (1, 1, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (2, 1, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (3, 1, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (10, 1, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (11, 1, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (12, 1, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (16, 2, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (21, 2, 1,
SYSDATE);
INSERT INTO melodie_apartine_playlist_user VALUES (24, 2, 1,
SYSDATE);
```

```
INSERT INTO melodie_apartine_playlist_user VALUES (6, 3, 2,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (7, 3, 2,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (24, 3, 2,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (2, 3, 2,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (21, 4, 4,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (22, 4, 4,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (23, 4, 4,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (8, 4, 4,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (9, 4, 4,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (13, 4, 4,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (4, 5, 5,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (5, 5, 5,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (15, 5, 5,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (16, 5, 5,  
SYSDATE);  
  
INSERT INTO melodie_apartine_playlist_user VALUES (17, 5, 5,  
SYSDATE);  
  
  
INSERT INTO utilizator_apreciaza_playlist VALUES(1, 3);  
INSERT INTO utilizator_apreciaza_playlist VALUES(1, 4);
```

```
INSERT INTO utilizator_apreciaza_playlist VALUES(1, 5);
INSERT INTO utilizator_apreciaza_playlist VALUES(5, 1);
INSERT INTO utilizator_apreciaza_playlist VALUES(5, 2);
INSERT INTO utilizator_apreciaza_playlist VALUES(5, 3);
INSERT INTO utilizator_apreciaza_playlist VALUES(4, 1);
INSERT INTO utilizator_apreciaza_playlist VALUES(3, 1);
INSERT INTO utilizator_apreciaza_playlist VALUES(3, 2);
INSERT INTO utilizator_apreciaza_playlist VALUES(3, 3);
```

```
alter session set nls_date_format = 'DD-MON-YYYY hh24:mi:ss';
```

```
INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 1, 3, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 2, 3, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 3, 3, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 4, 3, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 2, 7, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 3, 7, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 3, 9, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 3, 8, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 3, 10, SYSDATE);

INSERT INTO istoric_ascultare_podcast VALUES
(secv_id_asc_podcast.NEXTVAL, 2, 3, SYSDATE);
```

```
INSERT INTO istoric_ascultare_podcast VALUES  
(secv_id_asc_podcast.NEXTVAL, 4, 3, SYSDATE);  
  
INSERT INTO istoric_ascultare_podcast VALUES  
(secv_id_asc_podcast.NEXTVAL, 2, 7, SYSDATE);  
  
INSERT INTO istoric_ascultare_podcast VALUES  
(secv_id_asc_podcast.NEXTVAL, 3, 1, SYSDATE);  
  
INSERT INTO istoric_ascultare_podcast VALUES  
(secv_id_asc_podcast.NEXTVAL, 2, 1, SYSDATE);  
  
INSERT INTO istoric_ascultare_podcast VALUES  
(secv_id_asc_podcast.NEXTVAL, 3, 2, SYSDATE);  
  
INSERT INTO istoric_ascultare_podcast VALUES  
(secv_id_asc_podcast.NEXTVAL, 2, 2, SYSDATE);  
  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 6, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 6, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 6, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 6, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 5, 6, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 9, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 7, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 5, 11, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 11, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 2, SYSDATE);
```

```
INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 2, 3, SYSDATE);

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 4, SYSDATE);

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 2, 2, SYSDATE);

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 1, SYSDATE);

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('12-JAN-2022
14:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 16, TO_DATE('1-JUN-2022
14:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 5, TO_DATE('15-FEB-2022
18:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('5-MAR-2022
12:32:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('1-APR-2022
14:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 13, TO_DATE('4-APR-2022
14:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('1-MAY-2022
16:25:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 19, TO_DATE('4-JUN-2022
16:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 2, 5, TO_DATE('25-JUN-2022
18:35:27'));
```

```
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 5, TO_DATE('18-JUL-2022  
18:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 4, TO_DATE('4-APR-2022  
18:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('20-MAY-2022  
18:35:00'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 4, TO_DATE('15-JUN-2022  
18:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 8, TO_DATE('15-JUL-2022  
20:35:40'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 8, TO_DATE('16-SEP-2022  
10:53:25'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 8, TO_DATE('19-OCT-2022  
12:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 8, TO_DATE('12-NOV-2022  
18:20:30'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 4, TO_DATE('10-DEC-2022  
20:30:20'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 1, TO_DATE('12-JAN-2022  
16:45:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 12, TO_DATE('1-JUN-2022  
10:45:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 22, TO_DATE('15-FEB-2022  
12:35:27'));
```

```
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 23, TO_DATE('5-MAR-2022  
14:32:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 22, TO_DATE('1-APR-2022  
15:45:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 24, TO_DATE('4-APR-2022  
17:45:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 24, TO_DATE('1-MAY-2022  
19:25:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 19, TO_DATE('4-JUN-2022  
10:45:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 18, TO_DATE('25-JUN-2022  
12:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 18, TO_DATE('18-JUL-2022  
19:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 14, TO_DATE('4-APR-2022  
12:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 15, TO_DATE('20-MAY-2022  
11:35:00'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 14, TO_DATE('15-JUN-2022  
17:35:27'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 15, TO_DATE('15-JUL-2022  
22:35:40'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 12, TO_DATE('16-SEP-2022  
23:53:25'));
```

```
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 11, TO_DATE('12-NOV-2022  
21:20:30'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 9, TO_DATE('10-DEC-2022  
17:30:20'));  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 2, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 3, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 4, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 5, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 9, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 7, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 5, 13, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 4, 13, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 14, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 15, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 17, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 18, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 1, SYSDATE);  
  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('12-JAN-2022
```

```
14:45:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 16, TO_DATE('1-JUN-2022  
14:45:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 23, TO_DATE('15-FEB-2022  
18:35:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 22, TO_DATE('5-MAR-2022  
12:32:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 22, TO_DATE('1-APR-2022  
14:45:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 5, 6, TO_DATE('11-APR-2022  
15:30:20'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 13, TO_DATE('4-APR-2022  
14:45:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 4, TO_DATE('1-MAY-2022  
16:25:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 3, 19, TO_DATE('4-JUN-2022  
16:45:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 5, TO_DATE('25-JUN-2022  
18:35:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 5, TO_DATE('18-JUL-2022  
18:35:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 2, 8, TO_DATE('4-APR-2022  
18:35:27'));  
INSERT INTO istoric_ascultare_melodie VALUES  
(secv_id_asc_melodie.NEXTVAL, 1, 7, TO_DATE('20-MAY-2022
```

18:35:00'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 6, TO_DATE('15-JUN-2022
18:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 4, 4, TO_DATE('15-JUL-2022
20:35:40'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 8, TO_DATE('16-SEP-2022
10:53:25'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 8, TO_DATE('19-OCT-2022
12:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 8, TO_DATE('12-NOV-2022
18:20:30'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 4, 4, TO_DATE('10-DEC-2022
20:30:20'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 1, TO_DATE('12-JAN-2022
16:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 23, TO_DATE('1-JUN-2022
10:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 19, TO_DATE('6-JUN-2022
10:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 22, TO_DATE('15-FEB-2022
12:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 23, TO_DATE('5-MAR-2022
14:32:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 22, TO_DATE('1-APR-2022

15:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 24, TO_DATE('4-APR-2022
17:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 24, TO_DATE('1-MAY-2022
19:25:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 19, TO_DATE('4-JUN-2022
10:45:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 2, 18, TO_DATE('25-JUN-2022
12:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 2, 18, TO_DATE('18-JUL-2022
19:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 2, 14, TO_DATE('4-APR-2022
12:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 1, 15, TO_DATE('20-MAY-2022
11:35:00'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 14, TO_DATE('15-JUN-2022
17:35:27'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 4, 15, TO_DATE('15-JUL-2022
22:35:40'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 5, 6, TO_DATE('13-DEC-2022
15:30:20'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 12, TO_DATE('16-SEP-2022
23:53:25'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 3, 11, TO_DATE('12-NOV-2022

21:20:30'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 4, 16, TO_DATE('10-DEC-2022
17:30:20'));

INSERT INTO istoric_ascultare_melodie VALUES
(secv_id_asc_melodie.NEXTVAL, 5, 6, TO_DATE('11-DEC-2022
15:30:20'));

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în clauza FROM
- grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)
- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
- utilizarea a cel puțin 2 funcții pe siruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Ex 1.: Pentru fiecare melodie din baza de date să se afișeze durata, dacă aparține unui album sau este o melodie single și anul în care aceasta a fost lansată. Rezultatele vor fi ordonate descrescător după durata. Prin single se înțelege o melodie care nu face parte dintr-un album.

```
SELECT DISTINCT melodie.titlu, melodie.durata,  
DECODE(NVL(melodie.id_album, 0), 0, 'Este single', 'Apartine album')  
AS "Apartenenta album",
```

CASE

```
WHEN melodie.data_lansare IS NOT NULL THEN  
to_char(melodie.data_lansare, 'YYYY')
```

```
ELSE to_char(album.data_lansare, 'YYYY')
```

END AS "an lansare"

FROM melodie, album

```
WHERE melodie.id_album = album.id_album OR melodie.id_album IS  
NULL
```

ORDER BY durata DESC;

Elemente utilizate: NVL și DECODE în cadrul aceleiași cere + CASE + 1 funcție pe dată calendaristică

```
SELECT DISTINCT melodie.titlu, melodie.durata,
DECODE(NVL(melodie.id_album, 0), 0, 'Este single', 'Apartine album') AS "Aparține album",
CASE
    WHEN melodie.data_lansare IS NOT NULL THEN to_char(melodie.data_lansare, 'YYYY')
    ELSE to_char(album.data_lansare, 'YYYY')
END AS "an lansare"
FROM melodie, album
WHERE melodie.id_album = album.id_album OR melodie.id_album IS NULL
ORDER BY durata DESC;
```

Query Result | All Rows Fetched: 24 in 0.01 seconds

#	TITLU	DURATA	Aparține album	an lansare
1	Patek	280	Apartine album	2020
2	Lullaby	261	Apartine album	2020
3	Neon Guts	258	Apartine album	2017
4	444+222	247	Apartine album	2017
5	Tunechi	242	Apartine album	2020
6	Lacrimi	224	Apartine album	2020
7	wattba	221	Apartine album	2021
8	Orasul Pacatelor	214	Apartine album	2020
9	Vampiri	199	Apartine album	2020
10	rana	195	Apartine album	2021
11	Hell-O	194	Apartine album	2019
12	Plastic	193	Apartine album	2020
13	Teanc	189	Apartine album	2020
14	Nefiu	185	Este single	2018
15	Aripi	180	Apartine album	2020
16	Double Up	173	Este single	2020

Ex 2: Pentru fiecare utilizator al cărui nume are o lungime strict mai mare decât 5 și al cărui nume conține litera "a" (lower/upper case) să se afișeze câte melodii a ascultat fiecare. Un utilizator este luat în considerare doar dacă acesta a ascultat mai multe melodii decât utilizatorul care are cele mai puține melodii ascultate. Rezultatul urmează să fie sortat descrescător după numărul de melodii ascultate.

SELECT nume AS "Nume utilizator", playuri AS "Numar de melodii ascultate"

FROM utilizator,

SELECT utilizator.id_utilizator,

```

COUNT(istoric_ascultare_melodie.id_melodie) AS playuri
FROM istoric_ascultare_melodie, melodie, utilizator
WHERE melodie.id_melodie = istoric_ascultare_melodie.id_melodie
AND istoric_ascultare_melodie.id_utilizator = utilizator.id_utilizator
GROUP BY istoric_ascultare_melodie.id_utilizator
HAVING COUNT(istoric_ascultare_melodie.id_melodie)
> (SELECT MIN(COUNT(istoric_ascultare_melodie.id_melodie)) AS
playuri
FROM istoric_ascultare_melodie, melodie, utilizator
WHERE melodie.id_melodie =
istoric_ascultare_melodie.id_melodie AND
istoric_ascultare_melodie.id_utilizator = utilizator.id_utilizator
GROUP BY istoric_ascultare_melodie.id_utilizator)) aux
WHERE aux.id_utilizator = utilizator.id_utilizator AND
UPPER(utilizator.nume) LIKE '%A%' AND LENGTH(utilizator.nume) >
5
ORDER BY playuri DESC;

```

Elemente utilizate: subcerere nesincronizată în clauza FROM + grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri) + 2 funcții pe siruri de caractere.

Nume utilizator	Numar de melodiile ascultate
Marius	33
Zinggaaa	30
Tonut45Fotbal3	18

Ex 3: Pentru fiecare melodie lansată înainte de începerea anului 2020, să se afișeze numele acesteia și data ei de lansare.

WITH

aux AS (SELECT DISTINCT melodie.titlu,

CASE

WHEN melodie.data_lansare IS NOT NULL THEN
melodie.data_lansare

ELSE album.data_lansare

END AS lansare

FROM melodie, album

WHERE melodie.id_album = album.id_album OR
melodie.id_album IS NULL),

datacmp AS (SELECT TO_DATE('01-JAN-2020', 'DD-MON-YYYY') AS
datatocmp FROM dual)

SELECT titlu AS "TITLU MELODIE", lansare AS "DATA LANSARE"

FROM aux, datacmp

WHERE aux.lansare < datacmp.datatocmp

ORDER BY lansare;

Elemente utilizate: utilizarea cel puțin unui bloc de cerere (clauza WITH) + 1 funcție pe dată calendaristică

```
WITH
aux AS (SELECT DISTINCT melodie.titlu,
CASE
    WHEN melodie.data_lansare IS NOT NULL THEN melodie.data_lansare
    ELSE album.data_lansare
END AS lansare
FROM melodie, album
WHERE melodie.id_album = album.id_album OR melodie.id_album IS NULL),
datacmp AS (SELECT TO_DATE('01-JAN-2020', 'DD-MON-YYYY') AS datatocmp FROM dual)
SELECT titlu AS "TITLU MELODIE", lansare AS "DATA LANSARE"
FROM aux, datacmp
WHERE aux.lansare < datacmp.datatocmp
ORDER BY lansare;
```

Query Result



SQL

All Rows Fetched: 6 in 0.005 seconds

Ex 4: Să se afișeze în ordine lexicografică numele artiștilor care au cel puțin o melodie single. Prin single se înțelege o melodie care nu face parte dintr-un album.

```
SELECT nume AS "Nume artist"
FROM artist
WHERE EXISTS
    (SELECT artist_canta_melodie.id_melodie
     FROM artist_canta_melodie,
          (SELECT id_melodie iduri
           FROM melodie
           WHERE id_album IS NULL) singleuri
     WHERE id_melodie IN singleuri.iduri
       AND artist.id_artist = artist_canta_melodie.id_artist)
ORDER BY nume;
```

Elemente utilizate: subcerere sincronizată în care intervin cel puțin 3 tabele

```

SELECT nume AS "Nume artist"
FROM artist --- rezolva ce e mai sus
WHERE EXISTS
    (SELECT artist_canta_melodie.id_melodie
     FROM artist_canta_melodie,
          (SELECT id_melodie iduri
           FROM melodie
           WHERE id_album IS NULL) singleuri
     WHERE id_melodie IN singleuri.iduri
     AND artist.id_artist = artist_canta_melodie.id_artist)
ORDER BY nume;

```

Query Result

Nume artist
1 Amuly
2 Azteca
3 Calinacho
4 Ian
5 OSCAR

Ex 5: Pentru fiecare utilizator să se afișeze cea mai ascultată melodie a acestuia, alături de numărul de ascultări al melodiei de către acel utilizator. Rezultatul să fie sortat descrescător după numărul de ascultări.

```

SELECT utilizator.nume AS "Nume utilizator", melodie.titlu AS "Titlu
melodie", playuri AS "Numar ascultari"

FROM melodie, utilizator, (
    SELECT id_utilizator, id_melodie, playuri,
        row_number() OVER(PARTITION BY id_utilizator ORDER BY
playuri DESC) AS numar_rand
    FROM (
        SELECT utilizator.id_utilizator, melodie.id_melodie,
COUNT(istoric_ascultare_melodie.id_melodie) AS playuri
        FROM istoric_ascultare_melodie, melodie, utilizator

```

```

        WHERE melodie.id_melodie =
istoric_ascultare_melodie.id_melodie AND
istoric_ascultare_melodie.id_utilizator = utilizator.id_utilizator

        GROUP BY istoric_ascultare_melodie.id_melodie,
istoric_ascultare_melodie.id_utilizator

        ORDER BY playuri DESC)

) aux

```

```

WHERE aux.numar_rand = 1 AND melodie.id_melodie =
aux.id_melodie AND utilizator.id_utilizator = aux.id_utilizator

ORDER BY playuri DESC;

```

Element utilize: -

```

SELECT utilizator.nume AS "Nume utilizator", melodie.titlu AS "Titlu melodie", playuri AS "Numar ascultatori"
FROM melodie, utilizator, (
    SELECT id_utilizator, id_melodie, playuri,
    row_number() OVER(PARTITION BY id_utilizator ORDER BY playuri DESC) AS numar_rand
    FROM (
        SELECT utilizator.id_utilizator, melodie.id_melodie, COUNT(istoric_ascultare_melodie.id_melodie) AS playuri
        FROM istoric_ascultare_melodie, melodie, utilizator
        WHERE melodie.id_melodie = istoric_ascultare_melodie.id_melodie AND istoric_ascultare_melodie.id_utilizator = utilizator.id_utilizator
        GROUP BY istoric_ascultare_melodie.id_melodie, istoric_ascultare_melodie.id_utilizator
        ORDER BY playuri DESC)
    ) aux
    WHERE aux.numar_rand = 1 AND melodie.id_melodie = aux.id_melodie AND utilizator.id_utilizator = aux.id_utilizator
    ORDER BY playuri DESC;

```

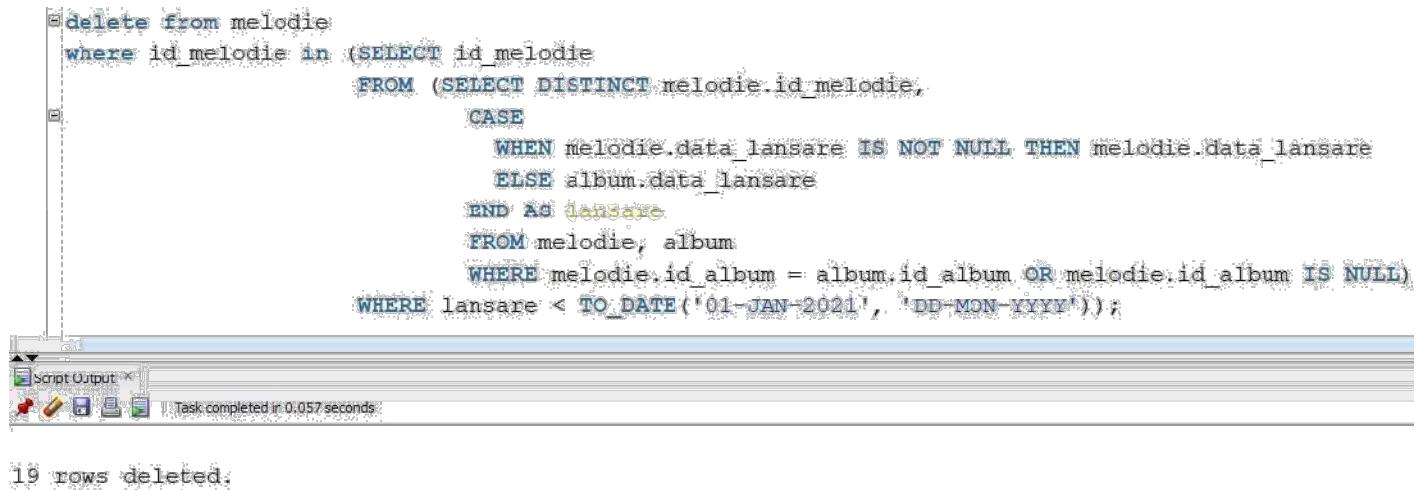
Query Result

Nume utilizator	Titlu melodie	Numar ascultatori
1ingeaus	444+222	8
1Marius	Biznis	6
2Ionut43Fotbal3	Oui, oui	5
3NelkaMM3	wattha	4
5JohnnytheBoy	Hell-O	3

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

Operația 1: Ștergeți din tabela MELODIE toate melodiile care au apărut înaintea anului 2021.

```
delete from melodie
where id_melodie in (SELECT id_melodie
                      FROM (SELECT DISTINCT melodie.id_melodie,
                                         CASE
                                             WHEN melodie.data_lansare IS NOT NULL
                                             THEN melodie.data_lansare
                                             ELSE album.data_lansare
                                         END AS lansare
                      FROM melodie, album
                     WHERE melodie.id_album = album.id_album OR
                           melodie.id_album IS NULL)
                     WHERE lansare < TO_DATE('01-JAN-2021',
                                           'DD-MON-YYYY'));
```



```
delete from melodie
where id_melodie in (SELECT id_melodie
                      FROM (SELECT DISTINCT melodie.id_melodie,
                                         CASE
                                             WHEN melodie.data_lansare IS NOT NULL THEN melodie.data_lansare
                                             ELSE album.data_lansare
                                         END AS lansare
                      FROM melodie, album
                     WHERE melodie.id_album = album.id_album OR melodie.id_album IS NULL)
                     WHERE lansare < TO_DATE('01-JAN-2021', 'DD-MON-YYYY'));
```

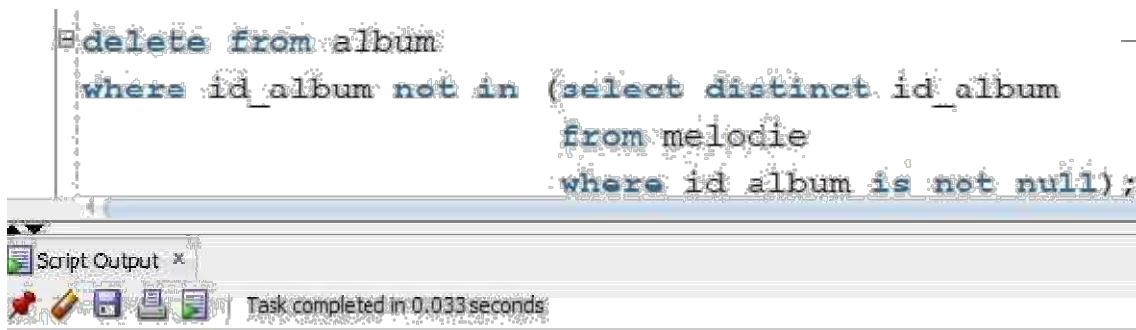
Script Output

Task completed in 0.057 seconds

19 rows deleted.

Operația 2: Ștergeți din tabela ALBUM toate albumele care nu conțin nicio melodie.

```
delete from album
where id_album not in (select distinct id_album
from melodie
where id_album is not null);
```



The screenshot shows a MySQL Workbench interface. In the top-left pane, there is a SQL editor window containing the following code:

```
delete from album
where id_album not in (select distinct id_album
from melodie
where id_album is not null);
```

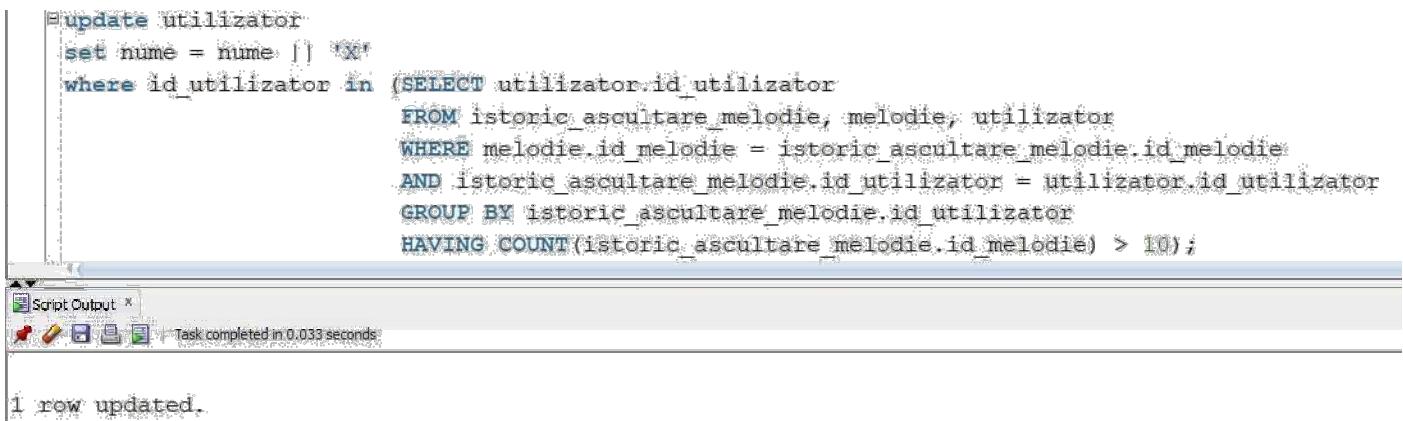
In the bottom-right pane, there is a "Script Output" window with the following message:

Task completed in 0.033seconds

6 rows deleted.

Operația 3: Adăugați un “X” la finalul numelor utilizatorilor care au ascultat mai mult de 10 melodii.

```
update utilizator
set nume = nume || 'X'
where id_utilizator in (SELECT utilizator.id_utilizator
                        FROM istoric_ascultare_melodie, melodie, utilizator
                        WHERE melodie.id_melodie =
                        istoric_ascultare_melodie.id_melodie
                        AND istoric_ascultare_melodie.id_utilizator =
                        utilizator.id_utilizator
                        GROUP BY istoric_ascultare_melodie.id_utilizator
                        HAVING
                        COUNT(istoric_ascultare_melodie.id_melodie) > 10);
```



The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the update query is typed. In the Script Output tab, the message "1 row updated." is displayed, indicating the query was successful. The status bar at the bottom shows "Task completed in 0.033 seconds".

```
update utilizator
set nume = nume || 'X'
where id_utilizator in (SELECT utilizator.id_utilizator
                        FROM istoric_ascultare_melodie, melodie, utilizator
                        WHERE melodie.id_melodie = istoric_ascultare_melodie.id_melodie
                        AND istoric_ascultare_melodie.id_utilizator = utilizator.id_utilizator
                        GROUP BY istoric_ascultare_melodie.id_utilizator
                        HAVING COUNT(istoric_ascultare_melodie.id_melodie) > 10);
```

1 row updated.

14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

Enunț: Creați o vizualizare care să conțină pentru fiecare utilizator: numele tuturor melodii ascultate de acesta alături de numărul de ascultări al melodiei respective.

```
CREATE VIEW vizualizare AS
    SELECT utilizator.nume AS "NUME UTILIZATOR", melodie.titlu
    AS "TITLU MELODIE",
    COUNT(istoric_ascultare_melodie.id_ascultare_melodie) AS
    "NUMAR PLAYURI"
    FROM utilizator, melodie, istoric_ascultare_melodie
    WHERE utilizator.id_utilizator =
    istoric_ascultare_melodie.id_utilizator
    AND istoric_ascultare_melodie.id_melodie = melodie.id_melodie
    GROUP BY utilizator.nume, melodie.titlu
    ORDER BY 3 DESC;
```

The screenshot shows a database interface with a script editor window containing the SQL code for creating a view. The code defines a view named 'vizualizare' that selects the user's name ('utilizator.nume'), the song title ('melodie.titlu'), and the count of plays ('COUNT(istoric_ascultare_melodie.id_ascultare_melodie)'). It joins three tables: 'utilizator', 'melodie', and 'istoric_ascultare_melodie'. The query filters by user ID and song ID, groups by user and song title, and orders by play count in descending order. Below the editor, a status bar indicates 'Task completed in 0.033 seconds'. At the bottom, a message says 'View VIZUALIZARE created.'

```
CREATE VIEW vizualizare AS
    SELECT utilizator.nume AS "NUME UTILIZATOR", melodie.titlu AS "TITLU MELODIE",
    COUNT(istoric_ascultare_melodie.id_ascultare_melodie) AS "NUMAR PLAYURI"
    FROM utilizator, melodie, istoric_ascultare_melodie
    WHERE utilizator.id_utilizator = istoric_ascultare_melodie.id_utilizator
    AND istoric_ascultare_melodie.id_melodie = melodie.id_melodie
    GROUP BY utilizator.nume, melodie.titlu
    ORDER BY 3 DESC;
```

Operație LMD permisă:

```
SELECT *
FROM vizualizare
WHERE "NUMAR PLAYURI" > 2;
```

The screenshot shows a MySQL Workbench interface with a query editor and a results table.

Query Editor Content:

```
SELECT *
FROM vizualizare
WHERE "NUMAR PLAYURI" > 2;
```

Results Table:

NUME UTILIZATOR	TITLU MELODIE	NUMAR PLAYURI
ingeaua	444+222	8
MariuS	Biznis	6
ingeaua	Double Up	6
Marius	Rocco	5
Ionut45Fotbal	Oui, oui	5
Ionut45Fotbal	Neon Guts	4
NelkaMM3	wattba	4
ingeaua	Tunechi	4
Marius	UFO	3
JohnnyTheBoy	444+222	3
JohnnyTheBoy	Helli-O	3
ingeaua	Never Recover	3
Marius	Teanc	3

Operație LMD neperemisă:

UPDATE vizualizare

SET "NUMAR PLAYURI" = 0

WHERE "NUMAR PLAYURI" < 3;

The screenshot shows a SQL query being run in Oracle SQL Developer. The query is:

```
UPDATE vizualizare
SET "NUMAR PLAYURI" = 0
WHERE "NUMAR PLAYURI" < 3;
```

The results pane displays the following error message:

```
Error starting at line : 1,058 in command -+
UPDATE vizualizare
SET "NUMAR PLAYURI" = 0
WHERE "NUMAR PLAYURI" < 3
Error at Command Line : 1,058 Column : 8
Error report -
SQL Error: ORA-01732: data manipulation operation not legal on this view
01732: 00000 - "data manipulation operation not legal on this view"
```

The interface includes tabs for "Query Result" and "Script Output". A status bar at the bottom indicates "Task completed in 0.033 seconds".

15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minim 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

Ex 1: Să se afișeze numele artiștilor care cântă toate melodiile lansate în anul 2018.

```
select nume
from artist
where id_artist in (
    with melodie_an as (
        SELECT DISTINCT melodie.id_melodie,
        CASE
            WHEN melodie.data_lansare IS NOT NULL THEN
                to_char(melodie.data_lansare, 'YYYY')
            ELSE to_char(album.data_lansare, 'YYYY')
        END AS an_lansare
        FROM melodie, album
        WHERE melodie.id_album = album.id_album OR
        melodie.id_album IS NULL
        ORDER BY an_lansare)
    select id_artist
    from artist_canta_melodie
    where id_melodie IN
        (select id_melodie
        from melodie_an
        where an_lansare = 2018)
    group by id_artist
```

```
having count(id_melodie) =  
    (select count(*)  
     from melodie_an  
     where an_lansare = 2018));
```

Element utilize: Division

Ex 2: Să se afișeze pentru fiecare utilizator născut înaintea anului 2003 sa se afiseze id-ul acestuia, numele, țara, cât și produsul cartezian dintre playlist-urile și albumele apreciate. Pentru playlist-uri și albume se vor afișa doar id-ul și numele acestora.

```
SELECT u.id_utilizator, u.nume, u.tara, p.id_playlist, p.nume,  
a.id_album, a.titlu  
FROM utilizator u  
FULL OUTER JOIN utilizator_apreciaza_playlist uap ON u.id_utilizator  
= uap.id_utilizator  
FULL OUTER JOIN playlist p ON uap.id_playlist = p.id_playlist  
FULL OUTER JOIN utilizator_apreciaza_album uaa ON u.id_utilizator  
= uaa.id_utilizator  
FULL OUTER JOIN album a ON uaa.id_album = a.id_album  
WHERE u.data_nastere <= TO_DATE('2003-01-01', 'YYYY-MM-DD');
```

Elemente utilizate: Outerjoin pe 4 tabele

Ex 3: Să se afișeze melodiile care se află în top 5 cele mai ascultate cât și numărul de ascultări ale acestora.

```
WITH statistica_melodii AS (
    SELECT melodie.titlu, COUNT(istoric_ascultare_melodie.id_melodie)
        AS playuri
    FROM istoric_ascultare_melodie, melodie
    WHERE melodie.id_melodie = istoric_ascultare_melodie.id_melodie
    AND id_album IS NOT NULL
    GROUP BY melodie.titlu
    ORDER BY playuri desc),
    top5 AS (SELECT playuri FROM statistica_melodii WHERE ROWNUM
    <= 5)
    SELECT *
    FROM statistica_melodii
    WHERE statistica_melodii.playuri in (SELECT * FROM top5);
```

Element utilizate: Analiza top n

16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.

Cerere:

Să se afișeze titlul, durata, genul și numele albumului pentru melodiile care sunt lansate înainte de 2021 și au o durată mai mare de 3 minute.

Pre-optimizare:

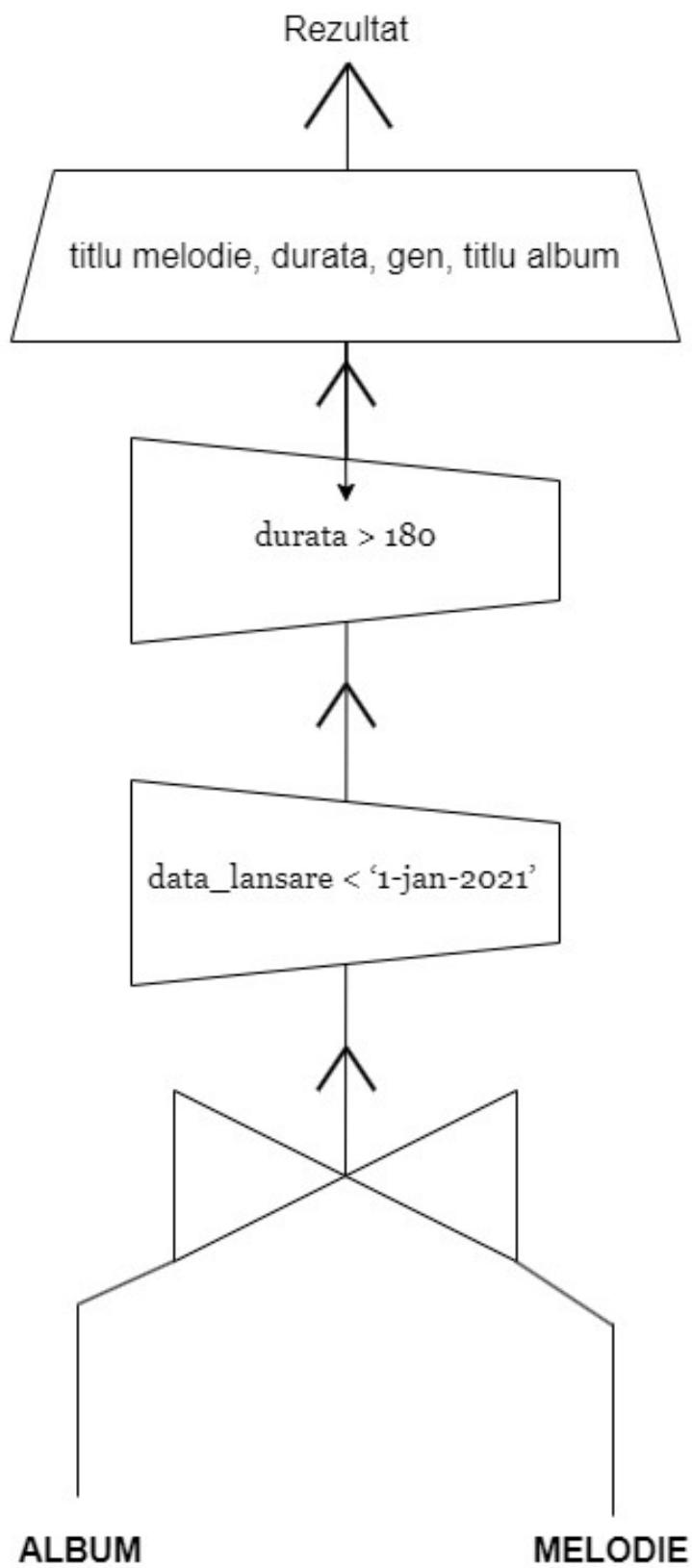
Cerere SQL:

```
SELECT melodie.titlu, durata, gen, album.titlu  
FROM melodie  
JOIN album ON album.id_album = melodie.id_album  
WHERE album.data_lansare < TO_DATE('01-JAN-2021', 'DD-MON-  
YYYY') AND melodie.durata > 180;
```

Expresie algebraică:

```
ρ1 = JOIN ( ALBUM, MELODIE )  
ρ2 = SELECT ( ρ1, data_lansare < '1-jan-2021' )  
ρ3 = SELECT ( ρ2, durata > 180 )  
ρ-final = PROJECT (ρ3, titlu melodie, durata, gen, titlu album)
```

Arbore algebric:



Post-optimizare:

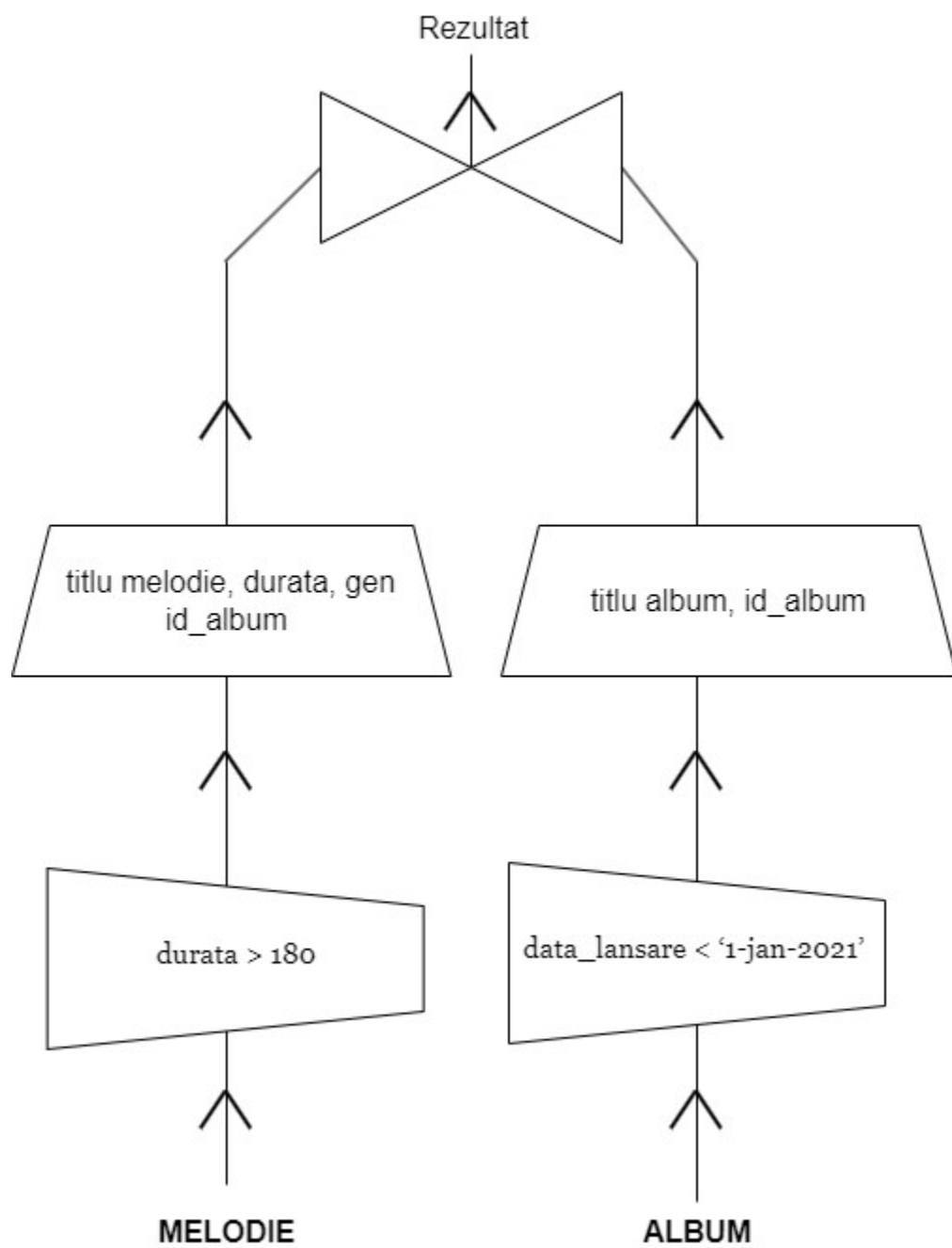
Cerere SQL:

```
SELECT melodie.titlu, durata, gen, album.titlu AS "Titlu album"
FROM (SELECT titlu, durata, gen, id_album
      FROM melodie WHERE durata > 180) melodie
JOIN (SELECT titlu, id_album
      FROM album WHERE data_lansare < TO_DATE ('01-JAN-2021',
'DD-MON-YYYY')) album
ON melodie.id_album = album.id_album;
```

Expresie algebrică:

```
 $\rho_1 = \text{SELECT ( MELODIE, durata > 180 )}$ 
 $\rho_2 = \text{PROJECT ( } \rho_1, \text{titlu, durata, gen, id\_album}$ 
 $\rho_3 = \text{SELECT ( ALBUM, data\_lansare < '1-jan-2021' )}$ 
 $\rho_4 = \text{PROJECT ( } \rho_3, \text{titlu, id\_album )}$ 
 $\rho\text{-final} = \text{JOIN ( } \rho_2, \rho_4 \text{ )}$ 
```

Arbore algebric:



- 17. a)Realizarea normalizării BCNF, FN4, FN5.
b)Aplicarea denormalizării, justificând necesitatea acesteia.**

a)

- Non BCNF:

TOP 3 MELODII ALE ANILOR			
AN_LANSARE	LOC_CLASAMENT	TITLU_MELODIE	AN_SI_LUNA_LANSARE
2019	1	Acasă	2019-Ian
2019	2	Apă	2019-Mai
2019	3	Minim doi	2019-Apr
2020	1	A ta	2020-Aug
2020	2	wattba	2020-Sep
2020	3	Nu ma uita	2020-Nov

- BCNF:

TOP 3 MELODII ALE ANILOR			
AN_LANSARE	LOC_CLASAMENT	TITLU_MELODIE	LUNA_LANSARE
2019	1	Acasă	Ianuarie
2019	2	Apă	Mai
2019	3	Minim doi	Aprilie
2020	1	A ta	August
2020	2	wattba	Septembrie
2020	3	Nu ma uita	Noiembrie

Un tabel se află în BCNF dacă acesta se află în FN3 și se elimină redundanțele datorate dependențelor funcționale. Se suprimă dependențele în care partea stângă nu este o supercheie.

Pentru exemplul de mai sus presupunem că atributul TITLU_MELODIE este unic în baza de date, astfel fiind și cheia noastră primară.

În tabelul nostru Non BCNF avem doar attribute prime întrucât cheile candidat sunt următoarele: {TITLU_MELODIE}, {LOC_CLASAMENT, AN_LANSARE}, {LOC_CLASAMENT, AN_SI_LUNA_LANSARE}

(AN_LANSARE, LOC_CLASAMENT)-> AN_SI_LUNA_LANSARE
 AN_SI_LUNA_LANSARE -> AN_LANSARE

Deci tabela noastră nu este în BCNF. Pentru a schimba acest lucru transformăm AN_SI_LUNA_LANSARE în LUNA_LANSARE pentru ca atributul AN_LANSARE să nu mai depindă de acesta.

- Non FN4:

ID_MELODIE	ID_ARTIST	ID_ALBUM

- FN4:

MELODIE_ARTIST		MELODIE_ALBUM	
ID_MELODIE	ID_ARTIST	MELODIE	ALBUM

Un tabel se află în FN4 dacă acesta se află în BCNF și elimină redundanțele datorate multidependenței. Se suprimă toate multidependențele care nu sunt și dependențe funcționale.

În exemplu avem drept cheie candidat (id_melodie, id_artist, id_album) și ca dependențe $\text{id_melodie} \rightarrow \text{id_artist}$, $\text{id_melodie} \rightarrow \text{id_album}$. Tabela fiind în BCNF rămâne să descompunem în două tabele: MELODIE_ARTIST (id_melodie, id_artist), MELODIE_ALBUM (id_melodie, id_album).

- Non FN5:

UTILIZATOR_APRECIAZA_PLAYLIST_USER		
ID_UTILIZATOR	ID_PLAYLIST	ID_DETINATOR_PLAYLIST

- FN5:

UTILIZATOR_APRECIAZA_PLAYLIST		UTILIZATOR_DETINE_PLAYLIST	
ID_UTILIZATOR	ID_PLAYLIST	ID_UTILIZATOR	ID_PLAYLIST

Un tabel se află în FN5 dacă acesta se află în FN4 și elimină redundanțele datorate dependenței ciclice. Se suprimă toate join-urile care nu sunt implicate de o cheie.

În exemplu avem drept cheie primară toate cele 3 id-uri. Deoarece tabelul poate fi spart în 2 fără ca informație să fie pierdută acesta nu se află în FN5. Spargem tabelul în alte două tabele care printr-un join duc la tabela inițială fără a se pierde informație sau a se adăuga redundanță.

b)

Denormalizarea constă în adăugare de date redundante pentru a crește viteza interogărilor la baza de date. De exemplu în proiectul realizat putem adăuga la o melodie numele albumului din care face parte pentru a evita join-ul repetat între tabelele ALBUM și MELODIE. Pentru a spori eficiența denormalizării aceasta trebuie realizată după sunt studiate tipurile de query-uri care se fac pe baza de date.