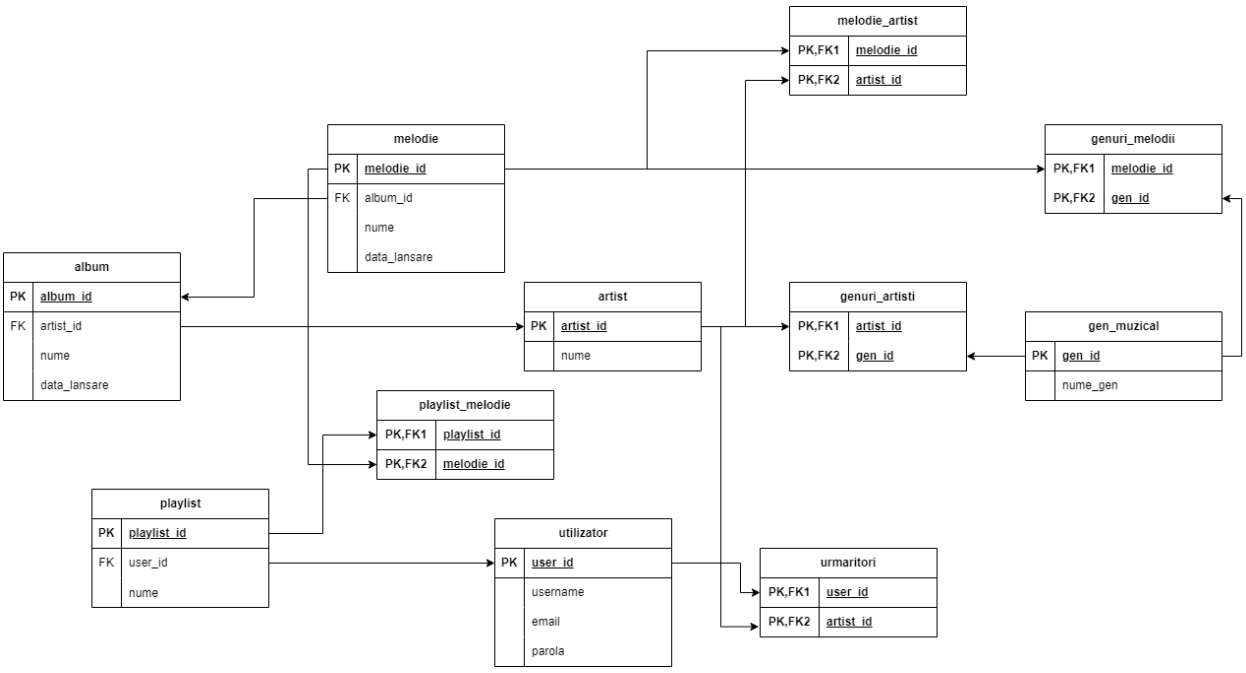
Problema gestiunii unei aplicații de muzică

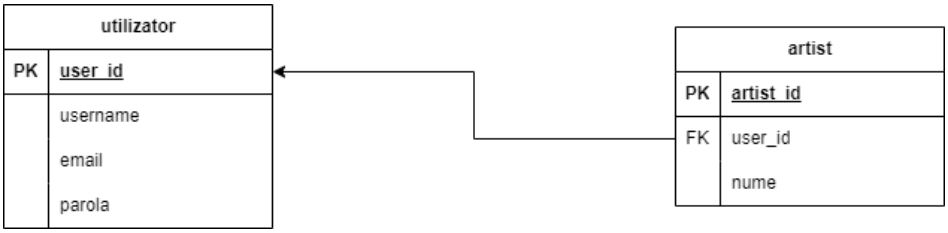


Pentru diagrama conceptuală de gestiune a unei aplicații de muzică rezolvați următoarele cerințe prin utilizarea unor pași de proiectare a unei baze de date relaționale – adăugarea de entități și (re)modelare, respectiv normalizare:

1. Determinați o relație de generalizare.
2. Puneți în evidență o relație în forma normală 3 și justificați de ce aceasta este in forma normală 3.
3. Adăugați la modelul existent posibilitatea ca un utilizator să își poată salva albumele selectate.
4. Scrieți interogarea SQL pentru afișarea tuturor melodiilor de pe albumul cu denumirea “Orasul 511”.
5. Scieți interogarea SQL pentru afișarea celui mai urmărit artist.
6. Scrieți interogarea SQL pentru a determina numele artistului care a scris melodia “eine Kleine Nachtmusik”.

Rezolvarea subiectului

1. O relație de generalizare posibilă este: “artistul este un utilizator”. Această relație de generalizare permite ca artistul să nu trebuiască să își facă un cont separat pentru a putea asculta la rândul său alți artiști.
   Pentru a face posibilă această relație, putem adăuga câmpul user_id în tabela artist, atributul având constrângerea de cheie străină cu referință la câmpul user_id din tabela utilizator.



2. Relația “un playlist conține una sau mai multe melodii” este in forma normală 3 deoarece respectă forma normală 2 si forma normală 1 și nu are dependențe tranzitive. Pentru a demonstra afirmația de mai sus, va trebui sa trecem prin toate cele 3 forme normale.
   Relația este in forma normală 1 deoarece fiecare atribut din relație este atomic. Adică, la intersecția de linii si coloane, avem o singură valoare care poate exista in tabel. Daca următorul tuplu ar fi fost posibil în relația prezentată, aceasta nu ar fi îndeplinit FN1 deoarece sunt inserate mai multe melodii intr-o singură casetă.

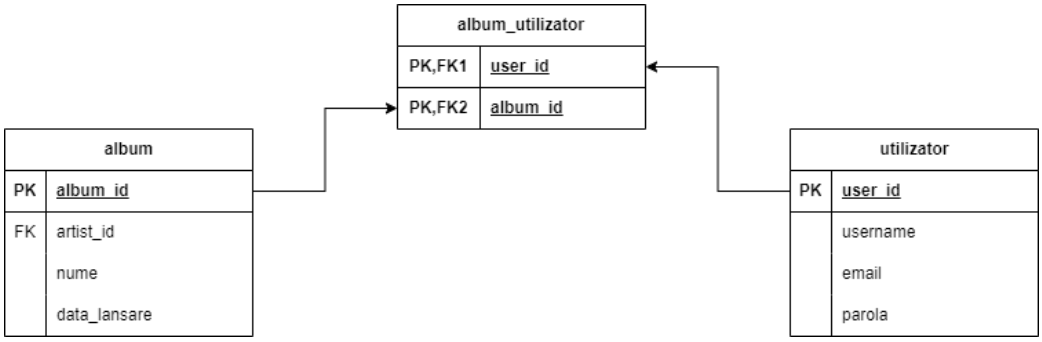
playlist_id	nume_utilizator	nume_melodie
0	Mihai	Du Hast, Engel, Zeit

   Relația este in forma normală 2 deoarece este in forma normală 1 si fiecare atribut este dependent functional de cheia primară. Observăm cum entitățile playlist si melodie sunt separate una de cealaltă, acestea fiind conectate printr-o tabelă de legatură care elimină dependențele dintre entități. Înainte ca relația să fie in forma normală 2, ar fi putut sa arate astfel:

playlist_id	user_id	nume_utilizator	melodie_id	album_id	nume_melodie	data_lansare
0	0	Mihai	0	0	Du Hast	1997-07-19
0	0	Mihai	1	0	Engel	1997-04-01
0	0	Mihai	2	1	Zeit	2022-04-29

Relația este in forma normală 3 deoarece este în forma normală 2 și nu conține dependențe tranzitive. Dacă tabela de legătură playlist_melodie ar fi avut atributul data_adaugare pentru a marca data în care melodia a fost adăugată în playlist și in plus am mai fi avut un atribut zile_adaugare care reține numărul de zile de când melodia a fost adăugată in playlist, relația nu ar mai fi respectat forma normală 3. Atributul zile_adaugare poate fi determinat automat din atributul data_adaugare.

3. Pentru a permite unui utilizator sa stocheze albumele selectate, trebuie creată o relație între album si utilizator, aceasta fiind: “Un utilizator poate salva unul sau mai multe albume”. Până acum relația este 1:N. Iar în sens invers, “un album poate fi salvat de unul sau mai mulți utilizatori”, rezultând relația M:N. Pentru a evita această situație, vom crea un tabel de legătură între utilizator și album pentru a păstra baza de date normalizată:



4. Pentru a determina care sunt melodiile care aparțin albumului specificat, vom împărți interogarea în doua părți:
   Prima data vom obține id-ul albumului cu denumirea ‘Orasul 511’:
   - select album_id from album where name = 'Orasul 511'

   Dupa care vom selecta numele melodiilor care au id-ul găsit anterior:

   - select nume from melodie where album_id = (select album_id from album where name = 'Orasul 511');

   Observăm utilizarea unei subinterogări pentru a delimita interogarea principală în cele două părți menționate mai sus.

5. Pentru a determina cel mai urmărit artist, trebuiesc unite tabelele artist și urmăritori printr-un inner join. Rezultatele trebuiesc grupate in funcție de artiști și ordonați descrescător în funcție de count-ul acestora. Cum s-a cerut doar cel mai urmărit artist, vom utiliza limit 1 pentru a returna doar primul tuplu.

- select artist.name, count(artist.artist_id) from artist inner join urmaritori on artist.artist_id = urmaritori.artist_id group by artist.artist_id order by count(artist.artist_id) desc limit 1;

6. Pentru a determina numele artistului care a scris melodia cu denumirea dată trebuie sa unim tabelele artist, melodie_artist si melodie prin inner join-uri. La finalul interogării, trebuie utilizată clauza WHERE pentru a selecta doar melodiile care au numele “eine Kleine Nachtmusik”.

   - select artist.name from artist
     inner join melodie_artist on melodie_artist.artist_id = artist.artist_id
     inner join melodie on melodie_artist.melodie_id = melodie.melodie_id
     where melodie.nume = 'eine Kleine Nachtmusik';