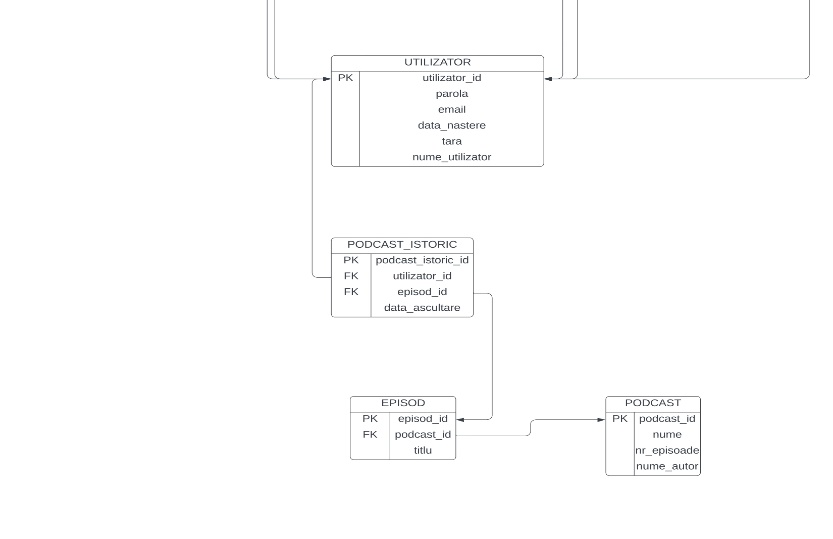
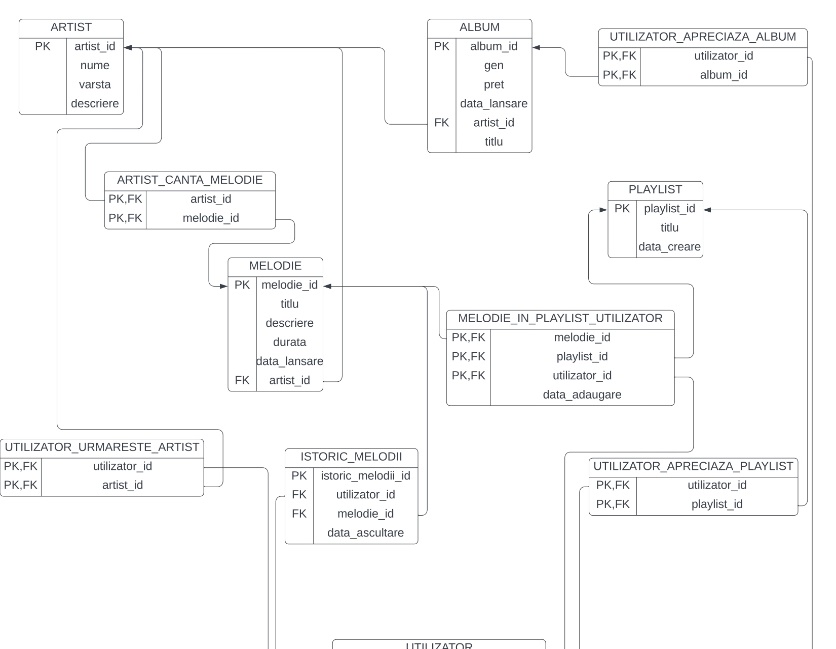
**Recapitulare 1**

1. Identificați diagrama conceptuală definită în proiectul prezentat la materia Baze de Date din anul
2. adăugați ca poză diagrama conceptuală identificată



1. adaptați cerințele exercițiilor 17 și 23 pentru diagrama conceptuală utilizată la punctul a (formulați cerința în limbaj natural, apoi rezolvați cererea propusă în SQL).

Enunț: Faceți o copie a tabelului artist și apoi generați un script de ștergere a acestuia.

SQL:

create table artist\_copy as (select \* from artist);

SPOOL ../sterg\_tabele\_output.txt

BEGIN

FOR t IN (SELECT table\_name FROM user\_tables WHERE table\_name = 'ARTIST\_COPY') LOOP

FOR c IN (SELECT constraint\_name FROM user\_constraints WHERE constraint\_type = 'R' AND table\_name = t.table\_name) LOOP

EXECUTE IMMEDIATE 'ALTER TABLE ' || t.table\_name || ' DROP CONSTRAINT ' || c.constraint\_name;

END LOOP;

EXECUTE IMMEDIATE 'DROP TABLE ' || t.table\_name;

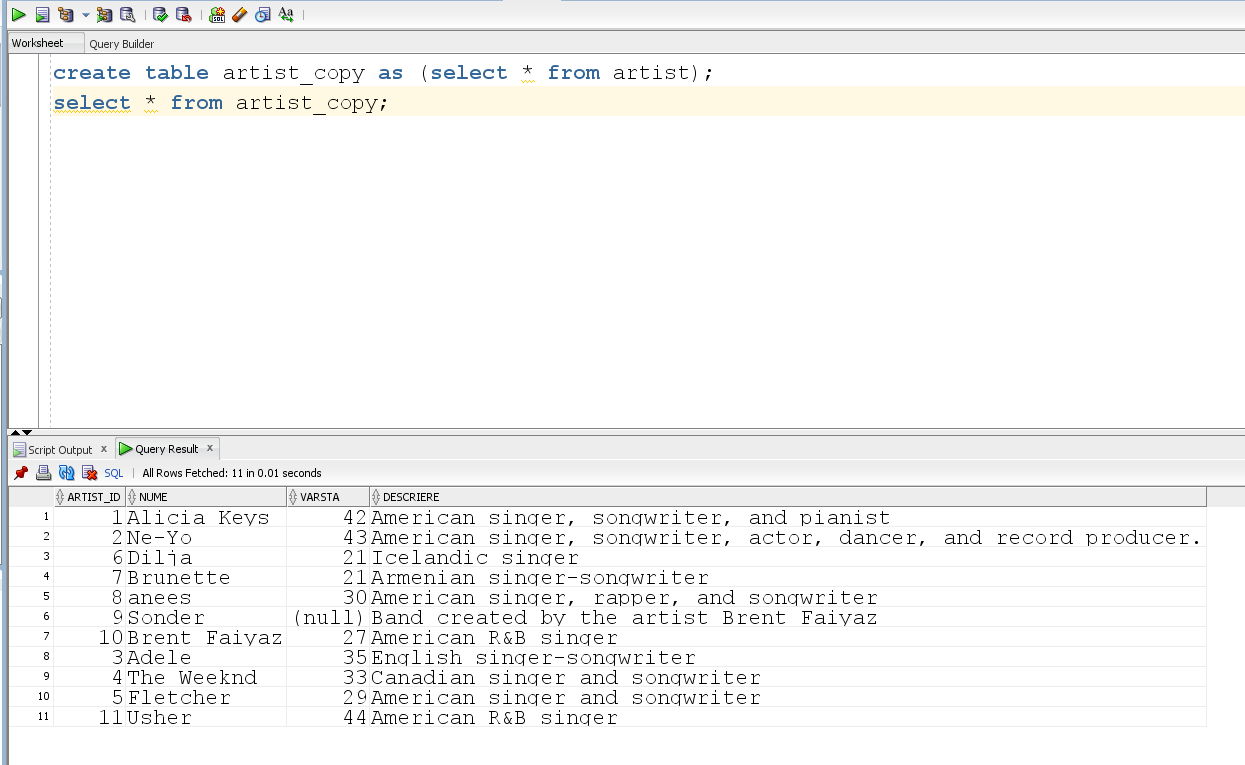
END LOOP;

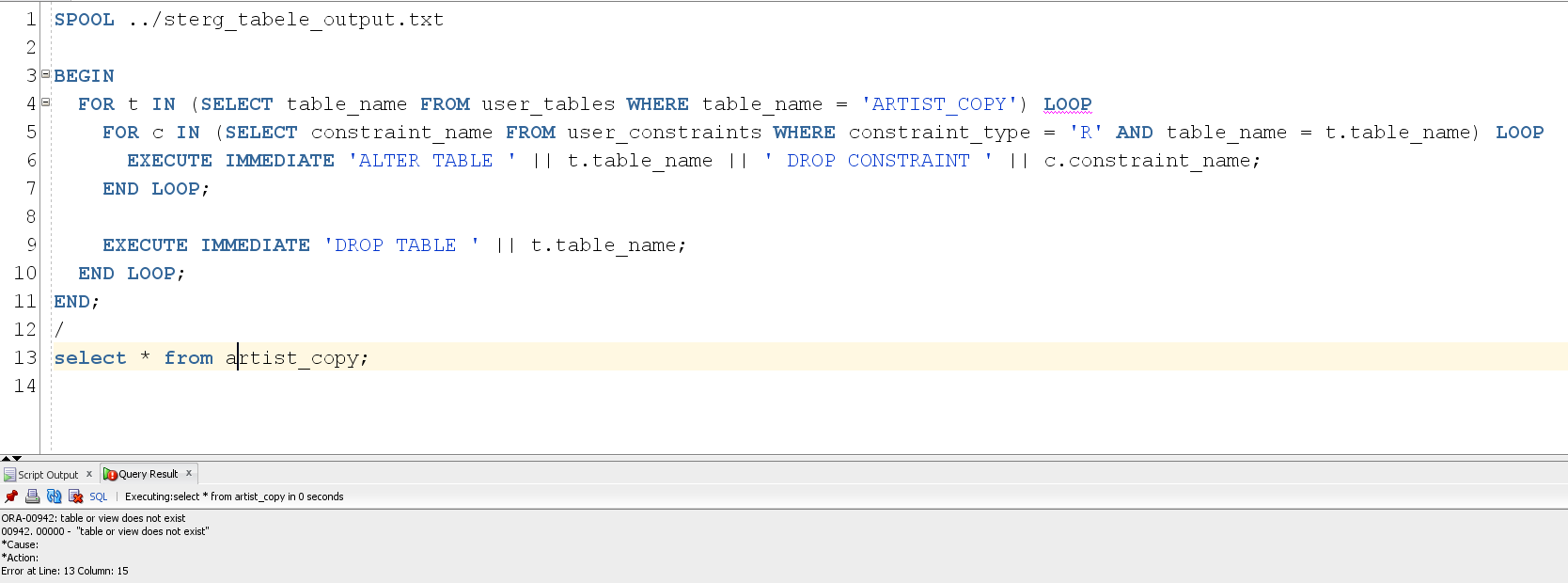
END;

/

Print screen-uri:







Enunt: Creati tabelul departments in baza dumneavoastra de date si apoi generati un script de inserare a valorilor pentru acesta.

Cod SQL:

CREATE TABLE departments (

department\_id NUMBER(4) PRIMARY KEY,

department\_name VARCHAR2(30),

manager\_id NUMBER(6),

location\_id NUMBER(4)

);

begin

for i in 1..5 loop

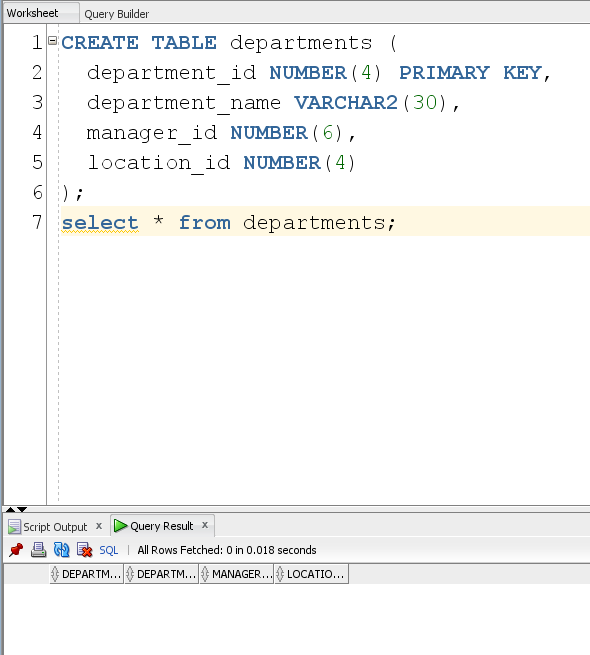
insert into departments (department\_id, department\_name, manager\_id, location\_id)

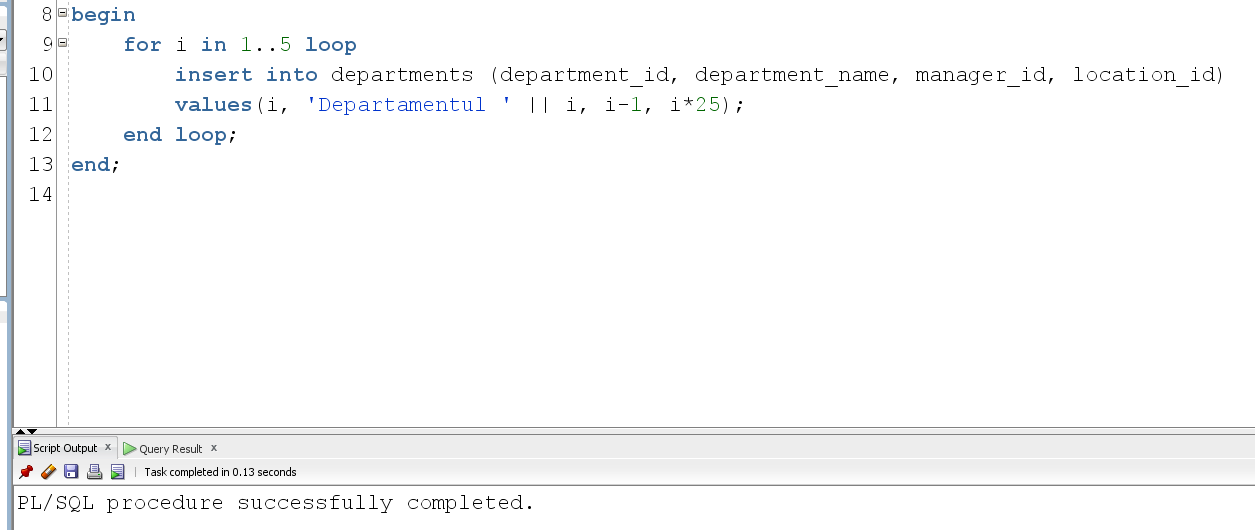
values(i, 'Departamentul ' || i, i-1, i\*25);

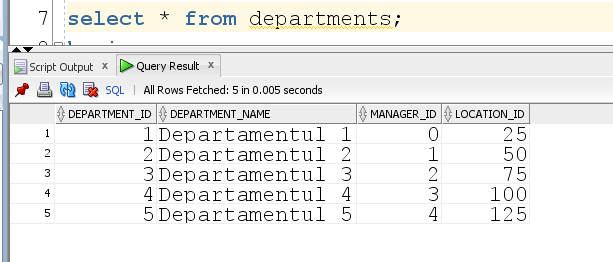
end loop;

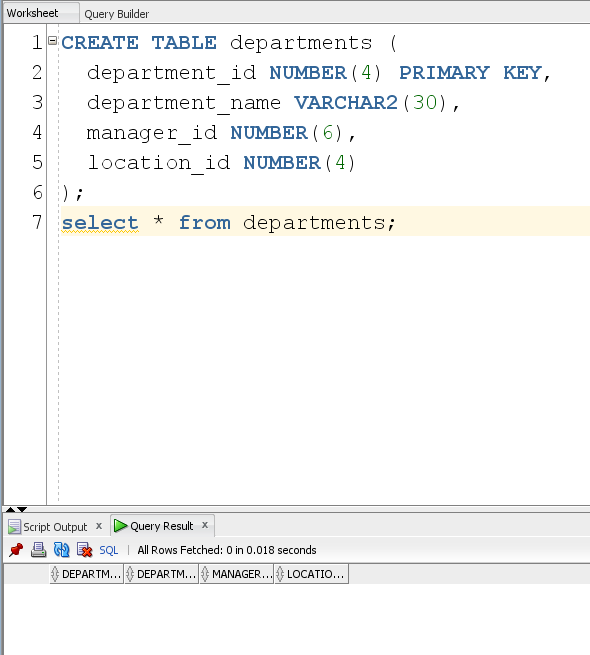
end;

Print screen-uri:









**Recapitulare 2**

9. De câte ori a împrumutat un membru (nume și prenume) fiecare film (titlu)?

Cod SQL:

with NrImpr as (

select

m.first\_name as "Nume",

m.last\_name as "Prenume",

t.title as "Titlu Film",

count(t.title\_id) as "Nr Imprumuturi"

from member m

join rental r on r.member\_id = m.member\_id

join title\_copy tc on tc.copy\_id = r.copy\_id and tc.title\_id = r.title\_id

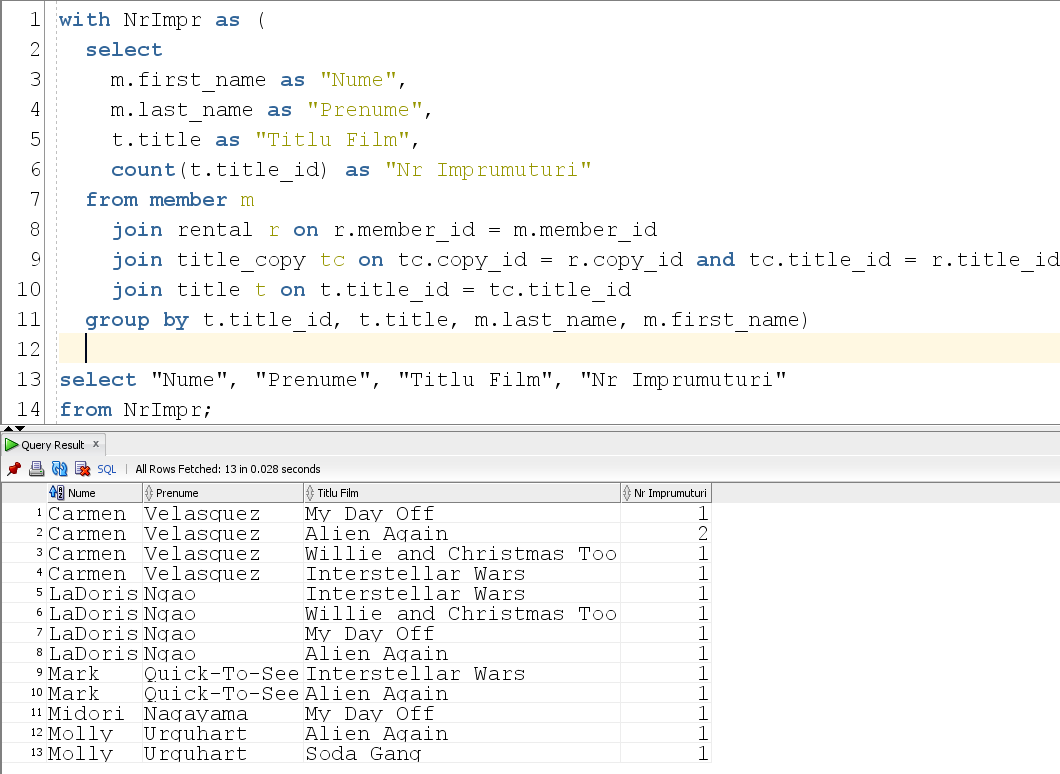
join title t on t.title\_id = tc.title\_id

group by t.title\_id, t.title, m.last\_name, m.first\_name)

select "Nume", "Prenume", "Titlu Film", "Nr Imprumuturi"

from NrImpr;

Print screen:



10. De câte ori a împrumutat un membru (nume și prenume) fiecare exemplar (cod) al unui film (titlu)?

Cod SQL:

with NrImpr as (

select m.last\_name as "Prenume", m.first\_name as "Nume", r.copy\_id as "Cod",

t.title as "Titlu Film", count(t.title\_id) as "Nr Imprumuturi"

from member m

join rental r on r.member\_id = m.member\_id

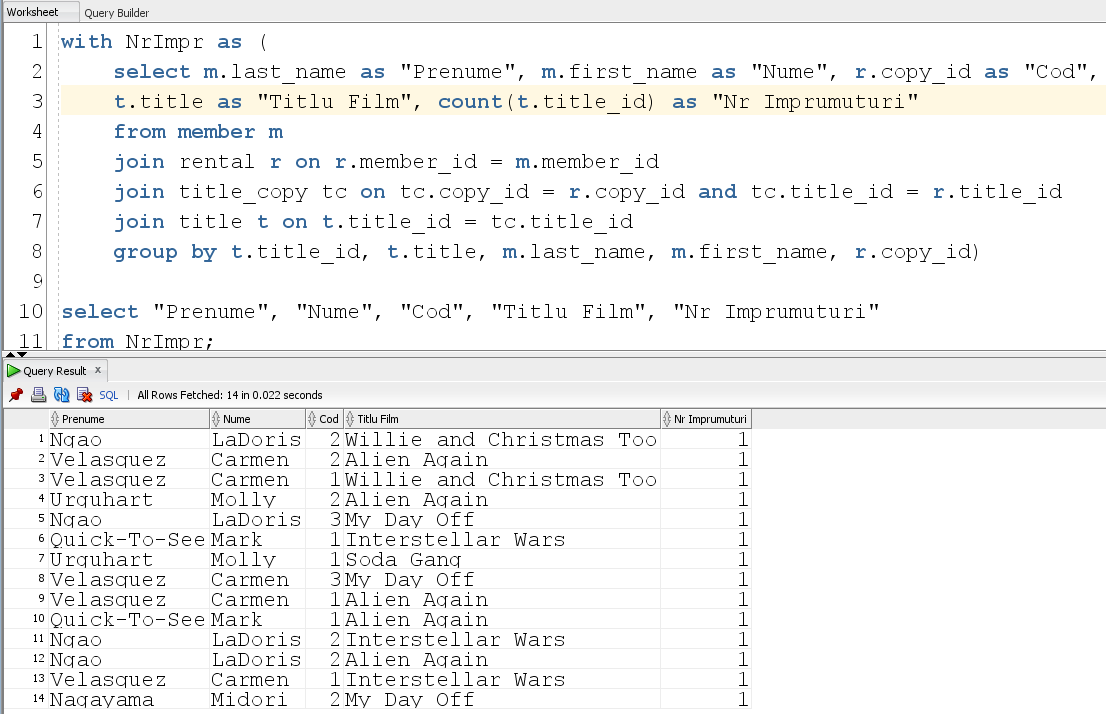
join title\_copy tc on tc.copy\_id = r.copy\_id and tc.title\_id = r.title\_id

join title t on t.title\_id = tc.title\_id

group by t.title\_id, t.title, m.last\_name, m.first\_name, r.copy\_id)

select "Prenume", "Nume", "Cod", "Titlu Film", "Nr Imprumuturi"

from NrImpr;

Print Screen: 

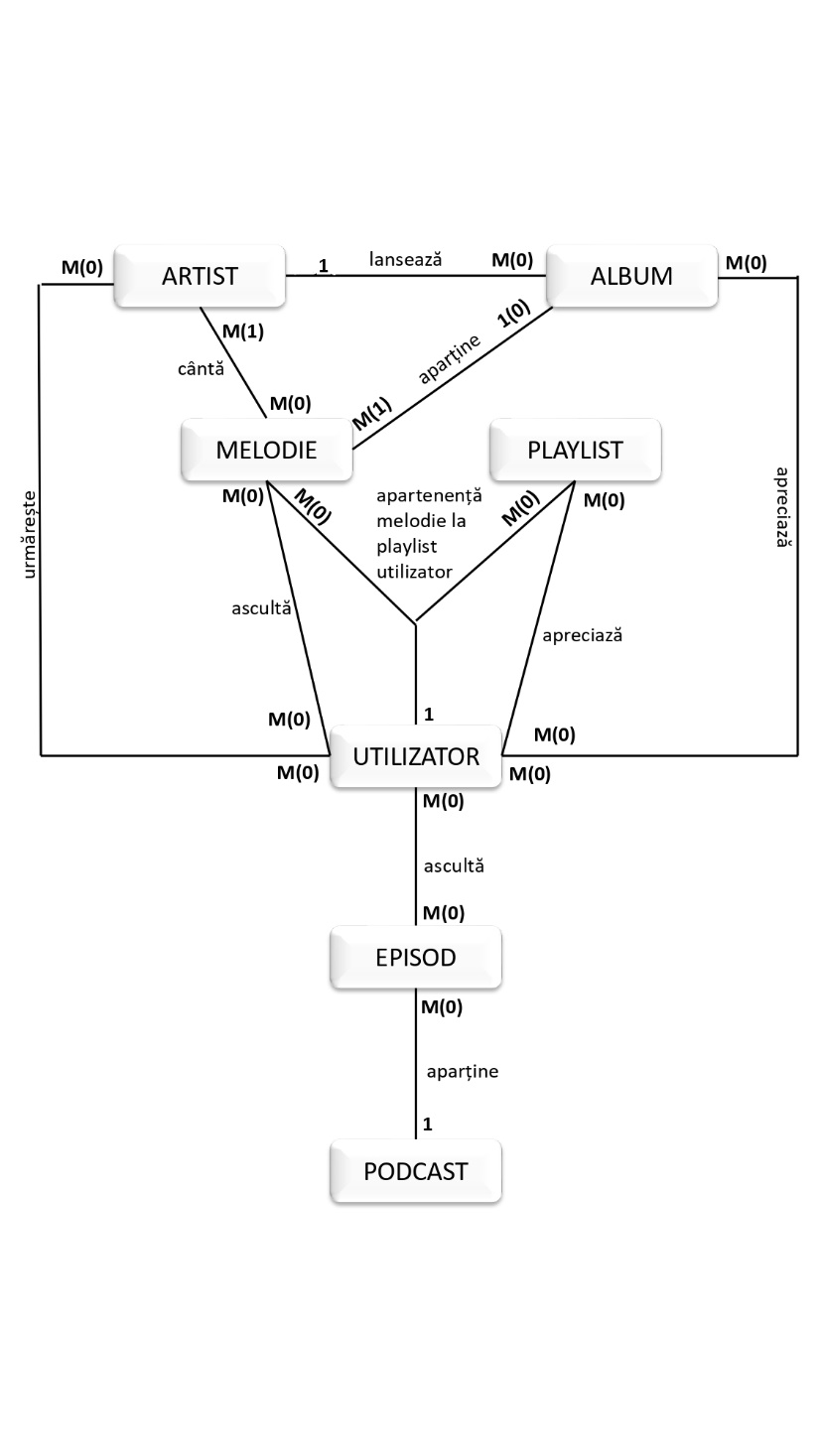
1. Identificați în diagrama Entitate-Relație utilizată în proiectul prezentat la materia Baze de Date din anul I, o relație din tip many-to-many.
2. **descrieți entitățile și relația dintre acestea:**

UTILIZATOR\_URMARESTE\_ARTIST: relatie de tip many-to-many care leaga entitatile UTILIZATOR si ARTIST si care exprima legatura dintre cele doua astfel: daca un utilizator urmareste un artist, va putea gasi mai usor lansarile (albumele si melodiile) acestuia. Relatia are cardinalitate minima 0:0 si cardinalitate maxima m:n.

UTILIZATOR: persoana care utilizeaza aceasta aplicatie prin intermediul contului pe care il creeaza. Cheia primara este utilizator\_id.

ARTIST: persoana care are posibilitatea de a incarca melodii si albume in aplicatie pentru a fi ascultate de utilizatori. Aceasta entitate are drept cheie primara artist\_id.

1. **realizați diagrama Entitate-Relație doar pentru această parte din proiect**

****

1. **descrieți modul de transformare al acestei relații în diagrama conceptuală, precizând toate cheile primare, cheile externe și alte atribute esențiale**

Entitatea UTILIZATOR are ca atribute:

→ utilizator\_id (**cheie primară**) = variabilă de tip întreg, care reprezintă id-ul unui utilizator.

→ nume\_utilizator = variabila de tip caracter, de lungime maximă 50, nenulă, unică, care reprezintă numele pe care ceilalți utilizatori îl văd în aplicație.

→ parola = variabilă de tip caracter, de lungime maximă 128, nenulă, care reprezintă parola cu care utilizatorul se autentifică în cont.

→ email = variabilă de tip caracter, de lungime maximă 150, care reprezintă adresa de e-mail a utilizatorului.

→data\_nastere = variabilă de tip dată calendaristică, nenulă, care reprezintă data nașterii utilizatorului.

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

Entitatea ARTIST are ca atribute:

→ artist\_id (**cheie primară**) = variabilă de tip întreg, care reprezintă id-ul unui artist.

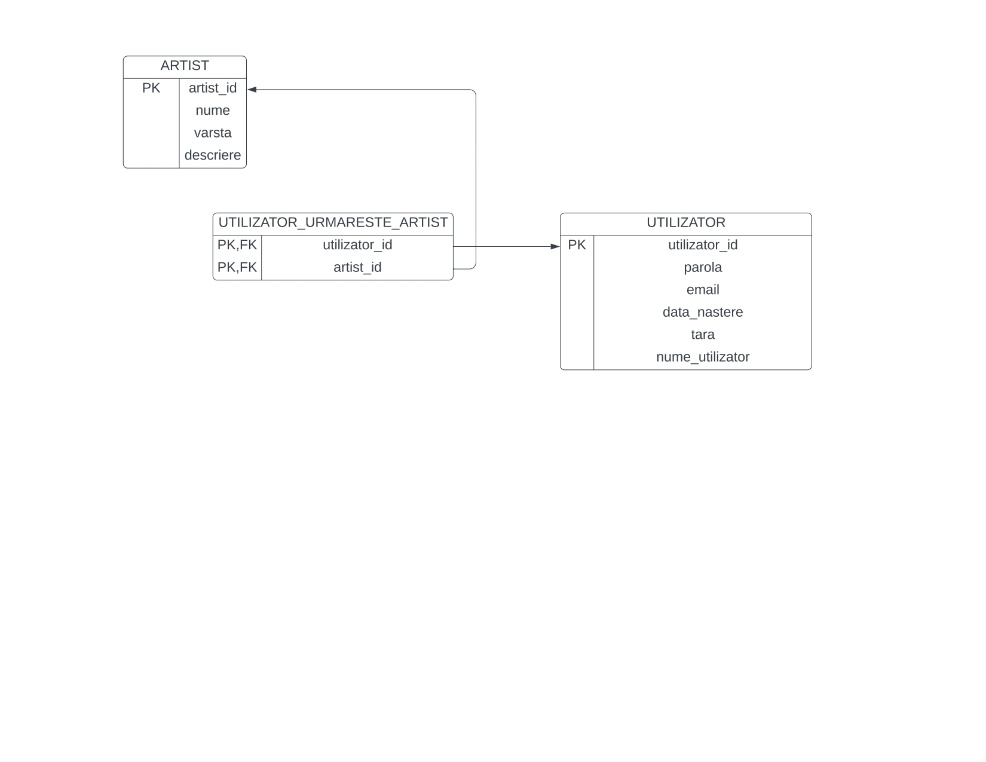
→ nume = variabilă de tip caracter, nenulă, de lungime maximă 100, care reprezintă numele artistului/trupei.

→ varsta = variabilă de tip întreg, care poate fi NULL pentru cazul trupelor, și reprezintă vârsta artistului, acolo unde e cazul.

→ descriere = variabilă de tip caracter, de dimensiune maximă 255, reprezintă o mică descriere pe care artistul/trupa o poate avea.

Relația UTILIZATOR\_urmărește\_ARTIST are ca atribute: utilizator\_id = variabilă de tip întreg, care reprezintă id-ul unui utilizator, atributul trebuie să corespundă unei chei primare din tabelul UTILIZATOR. artist\_id = variabilă de tip întreg, care reprezintă id-ul unui artist. Atributul trebuie să corespundă unei chei primare din tabelul ARTIST. Acest tabel present în diagrama conceptuală are drept **cheie primară**, cheia compusă din cele două attribute: utilizator\_id și artist\_id, unde acestea sunt și chei externe.

1. **realizați diagrama conceptuală doar pentru această parte din proiect**



1. **pe baza diagramei conceptuale de la punctul d, definiți în SQL tabelele și toate constrângerile necesare**

--tabel UTILIZATOR

CREATE TABLE UTILIZATOR(

utilizator\_id INT PRIMARY KEY,

nume\_utilizator VARCHAR(50) UNIQUE,

parola VARCHAR(128) NOT NULL,

email VARCHAR(150),

data\_nastere DATE NOT NULL,

tara VARCHAR(2) NOT NULL

);

--table ARTIST

CREATE TABLE ARTIST(

artist\_id INT PRIMARY KEY,

nume VARCHAR(100),

varsta INT,

descriere VARCHAR(255)

);

--tabel UTILIZATOR\_URMARESTE\_ARTIST

CREATE TABLE UTILIZATOR\_URMARESTE\_ARTIST(

utilizator\_id INT,

artist\_id INT,

PRIMARY KEY (utilizator\_id, artist\_id),

FOREIGN KEY(utilizator\_id) REFERENCES UTILIZATOR(utilizator\_id),

FOREIGN KEY(artist\_id) REFERENCES ARTIST(artist\_id)

);

1. **Care este id-ul si numele celui mai urmarit artist? Afisati si numarul de urmariri**

Cod sql: select a.artist\_id, a.nume, count(uua.utilizator\_id) as nr\_urm

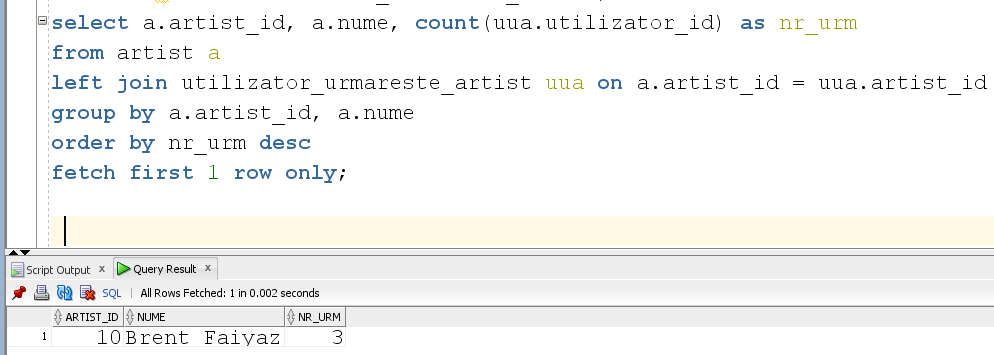
from artist a

left join utilizator\_urmareste\_artist uua on a.artist\_id = uua.artist\_id

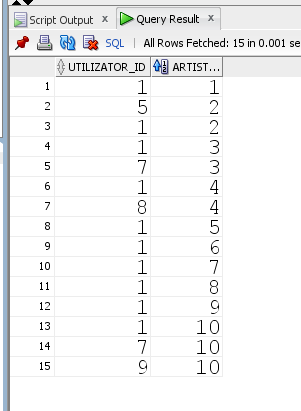
group by a.artist\_id, a.nume

order by nr\_urm desc

fetch first 1 row only;

Print screen: 

Continutul tabelului utilizator\_urmareste\_artist pentru verificare:



**OBS** nu stiam daca cerinta este o interpretare potrivita pentru adaptarea unui enunt de la 4-12, initial am facut altul dar nu pentru tabelele prezentate in diagrama de la punctul d).

Enunt: Aflati cate melodii apartin celui mai ascultat gen musical.

Cod sql:

with nrcantece as (

select a.gen, count(ist.melodie\_id) as nr\_cantece

from istoric\_melodii ist

join melodie m on ist.melodie\_id = m.melodie\_id

join album a on m.album\_id=a.album\_id

group by a.gen

)

select gen, nr\_cantece

from nrcantece

where nr\_cantece = (

select max(nr\_cantece)

from nrcantece);

