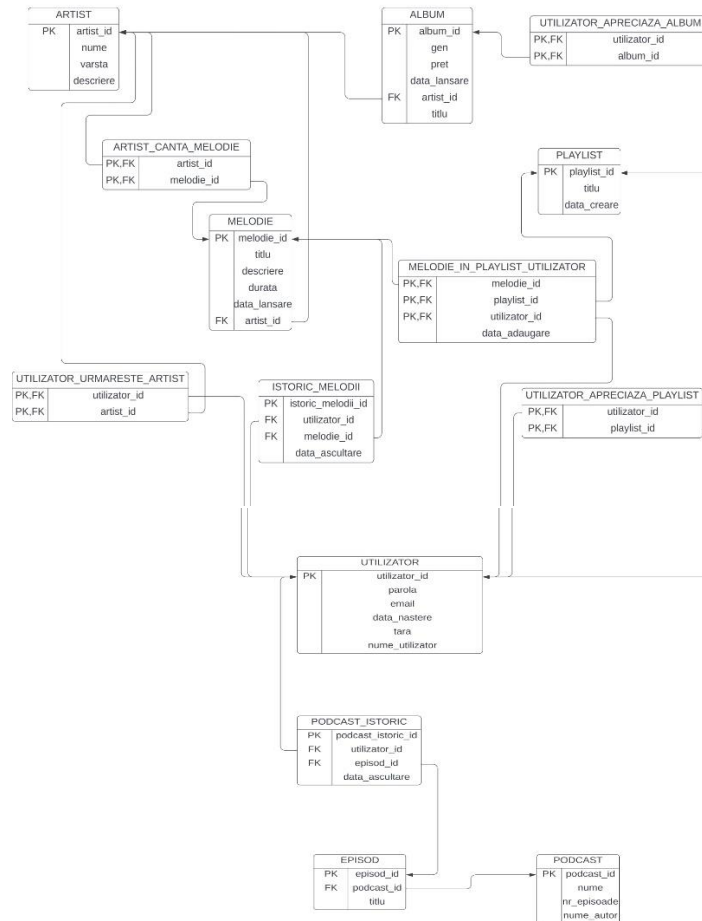


## Recapitulare 1

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



- b. 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:

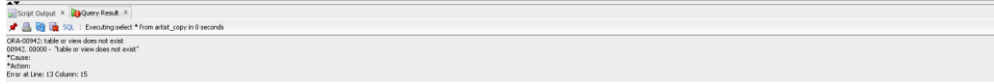


Table ARTIST\_COPY created.

The screenshot shows the same SQL query editor as before, but now the query result is displayed below the editor. The result is a table with 11 rows and 4 columns: ARTIST\_ID, NAME, VARSTA, and DESCRIERE.

	ARTIST_ID	NAME	VARSTA	DESCRIERE
1	1	Alicia Keys	42	American singer, songwriter, and pianist
2	2	Ne-Yo	43	American singer, songwriter, actor, dancer, and record producer.
3	6	Dilija	21	Icelandic singer
4	7	Brunette	21	Armenian singer-songwriter
5	8	anees	30	American singer, rapper, and songwriter
6	9	Sonder	(null)	Band created by the artist Brent Faiyaz
7	10	Brent Faiyaz	27	American R&B singer
8	3	Adele	35	English singer-songwriter
9	4	The Weeknd	33	Canadian singer and songwriter
10	5	Fletcher	29	American singer and songwriter
11	11	Usher	44	American R&B singer

```
1 SPOOL ../sterg_tabele_output.txt
2
3 BEGIN
4 FOR t IN (SELECT table_name FROM user_tables WHERE table_name = 'ARTIST_COPY') LOOP
5   FOR c IN (SELECT constraint_name FROM user_constraints WHERE constraint_type = 'R' AND table_name = t.table_name) LOOP
6     EXECUTE IMMEDIATE 'ALTER TABLE ' || t.table_name || ' DROP CONSTRAINT ' || c.constraint_name;
7   END LOOP;
8
9   EXECUTE IMMEDIATE 'DROP TABLE ' || t.table_name;
10 END LOOP;
11 END;
12 /
13 select * from artist_copy;
14
```



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:

```
Worksheet Query Builder
1 CREATE TABLE departments (
2   department_id NUMBER(4) PRIMARY KEY,
3   department_name VARCHAR2(30),
4   manager_id NUMBER(6),
5   location_id NUMBER(4)
6 );
7 select * from departments;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 0 in 0.018 seconds  
DEPARTM... DEPARTM... MANAGER... LOCATIO...

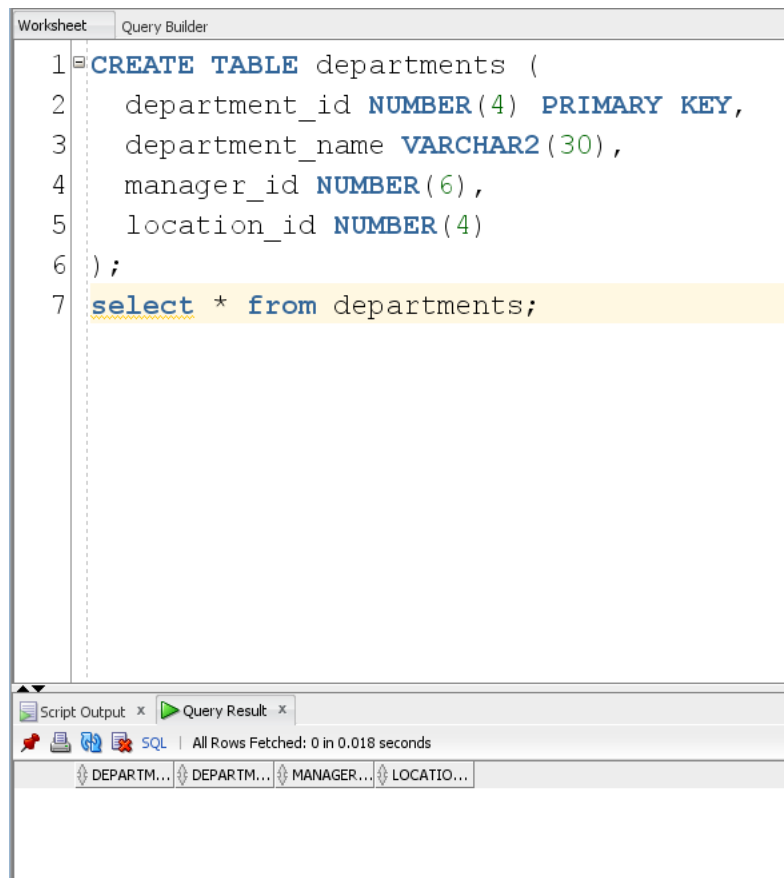
```
8 begin
9   for i in 1..5 loop
10     insert into departments (department_id, department_name, manager_id, location_id)
11       values(i, 'Departamentul ' || i, i-1, i*25);
12   end loop;
13 end;
14
```

Script Output x Query Result x  
Task completed in 0.13 seconds  
PL/SQL procedure successfully completed.

```
7 select * from departments;
```

Script Output x Query Result x  
SQL | All Rows Fetched: 5 in 0.005 seconds

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	1	Departamentul 1	0	25
2	2	Departamentul 2	1	50
3	3	Departamentul 3	2	75
4	4	Departamentul 4	3	100
5	5	Departamentul 5	4	125



## 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:

```
1 with NrImpr as (  
2   select  
3     m.first_name as "Nume",  
4     m.last_name as "Prenume",  
5     t.title as "Titlu Film",  
6     count(t.title_id) as "Nr Imprumuturi"  
7   from member m  
8     join rental r on r.member_id = m.member_id  
9     join title_copy tc on tc.copy_id = r.copy_id and tc.title_id = r.title_id  
10    join title t on t.title_id = tc.title_id  
11   group by t.title_id, t.title, m.last_name, m.first_name)  
12  
13 select "Nume", "Prenume", "Titlu Film", "Nr Imprumuturi"  
14 from NrImpr;
```

Query Result x All Rows Fetched: 13 in 0.028 seconds

	Nume	Prenume	Titlu Film	Nr Imprumuturi
1	Carmen	Velasquez	My Day Off	1
2	Carmen	Velasquez	Alien Again	2
3	Carmen	Velasquez	Willie and Christmas Too	1
4	Carmen	Velasquez	Interstellar Wars	1
5	LaDoris	Ngao	Interstellar Wars	1
6	LaDoris	Ngao	Willie and Christmas Too	1
7	LaDoris	Ngao	My Day Off	1
8	LaDoris	Ngao	Alien Again	1
9	Mark	Quick-To-See	Interstellar Wars	1
10	Mark	Quick-To-See	Alien Again	1
11	Midori	Nagayama	My Day Off	1
12	Molly	Urquhart	Alien Again	1
13	Molly	Urquhart	Soda Gang	1

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:

Worksheet

Query Builder

```
1 with NrImpr as (
2     select m.last_name as "Prenume", m.first_name as "Nume", r.copy_id as "Cod",
3     t.title as "Titlu Film", count(t.title_id) as "Nr Imprumuturi"
4     from member m
5     join rental r on r.member_id = m.member_id
6     join title_copy tc on tc.copy_id = r.copy_id and tc.title_id = r.title_id
7     join title t on t.title_id = tc.title_id
8     group by t.title_id, t.title, m.last_name, m.first_name, r.copy_id)
9
10 select "Prenume", "Nume", "Cod", "Titlu Film", "Nr Imprumuturi"
11 from NrImpr;
```

Query Result x

All Rows Fetched: 14 in 0.022 seconds

	Prenume	Nume	Cod	Titlu Film	Nr Imprumuturi
1	Ngao	LaDoris	2	Willie and Christmas Too	1
2	Velasquez	Carmen	2	Alien Again	1
3	Velasquez	Carmen	1	Willie and Christmas Too	1
4	Urquhart	Molly	2	Alien Again	1
5	Ngao	LaDoris	3	My Day Off	1
6	Quick-To-See	Mark	1	Interstellar Wars	1
7	Urquhart	Molly	1	Soda Gang	1
8	Velasquez	Carmen	3	My Day Off	1
9	Velasquez	Carmen	1	Alien Again	1
10	Quick-To-See	Mark	1	Alien Again	1
11	Ngao	LaDoris	2	Interstellar Wars	1
12	Ngao	LaDoris	2	Alien Again	1
13	Velasquez	Carmen	1	Interstellar Wars	1
14	Nagayama	Midori	2	My Day Off	1

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.

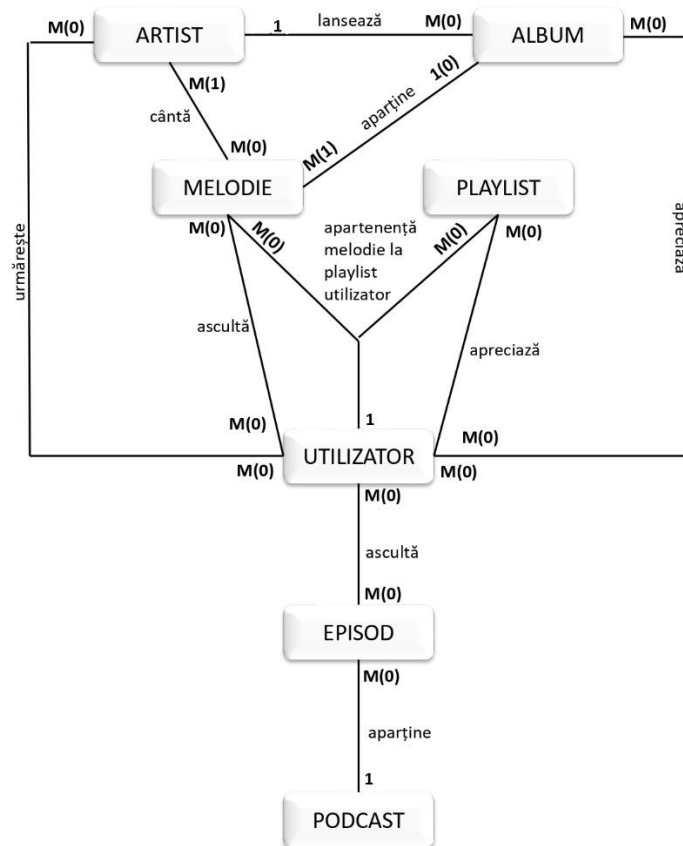
**a. descrieți entitățile și relația dintre acestea:**

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.

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





**c. 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:

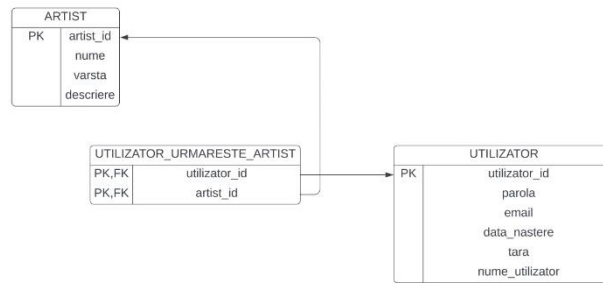
- utilizator\_id (**cheie primară**) = variabilă de tip întreg, care reprezintă id-ul unui utilizator.
- nume\_utilizator = variabila de tip caracter, de lungime maximă 50, nenulă, unică, care reprezintă numele pe care ceilalți utilizatori îl văd în aplicație.
- parola = variabilă de tip caracter, de 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 prezent în diagrama conceptuală are drept **cheie primară**, cheia compusă din cele două atribute: utilizator\_id și artist\_id, unde acestea sunt și chei externe.

d. realizați diagrama conceptuală doar pentru această parte din proiect



e. 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)

);

```

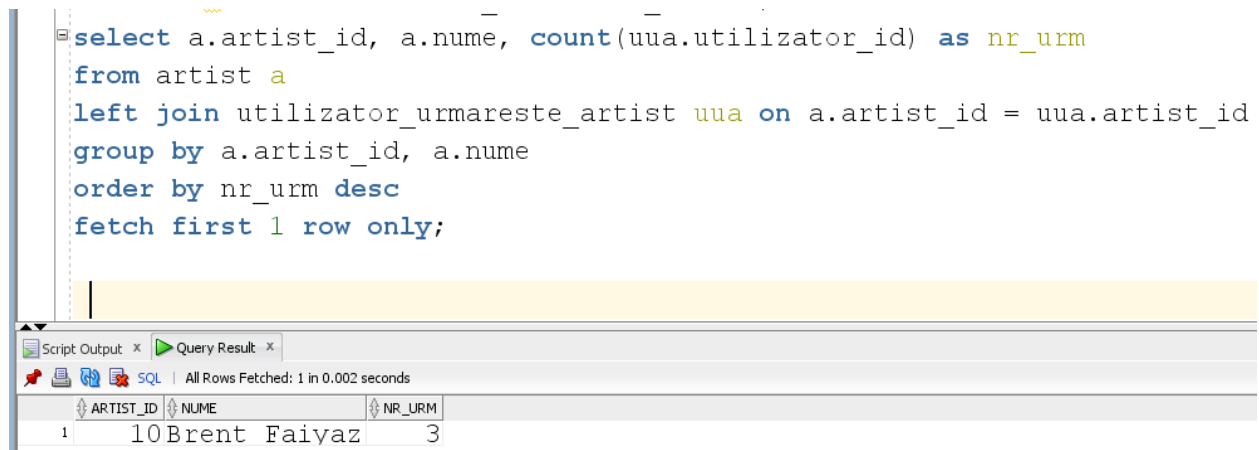
**f. Care este id-ul si numele celui mai urmarit artist? Afisati si numarul de urmariri**

```

Cod sql: select a.artist_id, a.numa, count(uua.utilizator_id) as nr_urm
from artist a
left join utilizator_urmaresta_artist uua on a.artist_id = uua.artist_id
group by a.artist_id, a.numa
order by nr_urm desc
fetch first 1 row only;

```

Print screen:



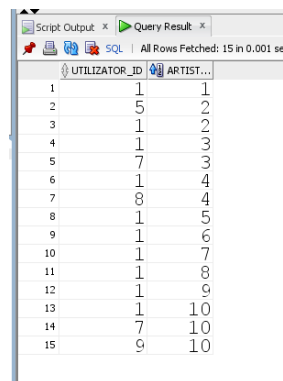
```

select a.artist_id, a.numa, count(uua.utilizator_id) as nr_urm
from artist a
left join utilizator_urmaresta_artist uua on a.artist_id = uua.artist_id
group by a.artist_id, a.numa
order by nr_urm desc
fetch first 1 row only;

```

ARTIST_ID	NUMA	NR_URM
10	Brent Faiyaz	3

Continutul tabelului utilizator\_urmaresta\_artist pentru verificare:



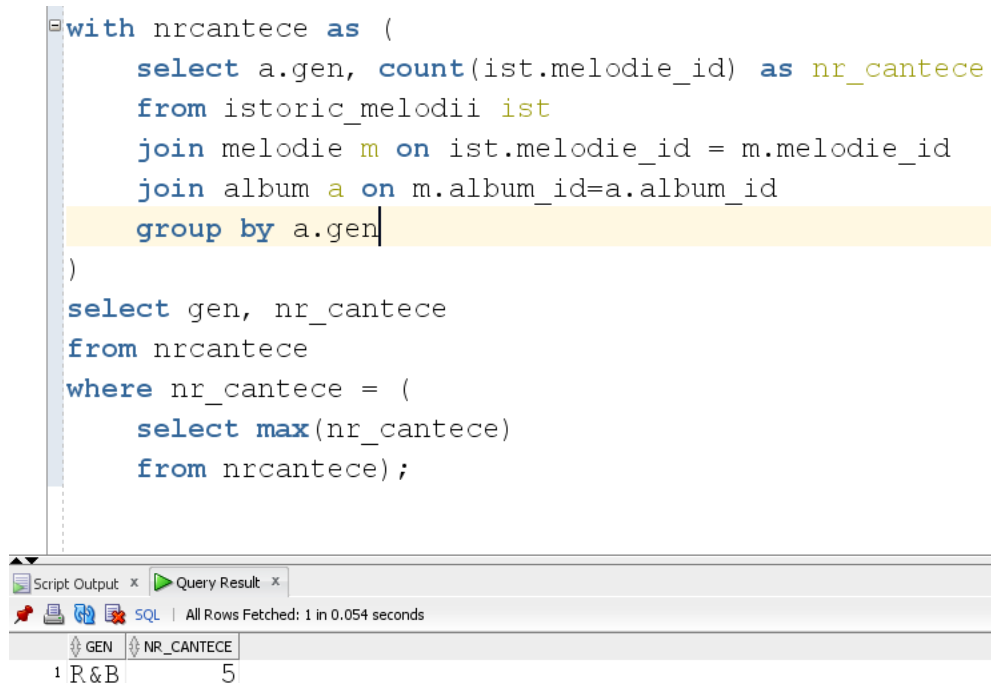
UTILIZATOR_ID	ARTIST_ID
1	1
5	2
1	2
1	3
7	3
1	4
8	4
1	5
1	6
1	7
1	8
1	9
1	10
7	10
9	10

**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);
```



The screenshot shows a SQL IDE with a query editor and a results pane. The query is the same as the one provided in the previous block. The results pane shows a single row with the genre 'R&B' and the count '5'.

```
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);
```

GEN	NR_CANTECE
1 R&B	5