

Şerba Ioana-Raluca  
Grupa 141

# Proiect baze de date Gestionarea unei case de discuri

Proiect realizat de Şerba Ioana-Raluca  
Grupa 141

## Cuprins

.....	1
<b>1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare ...</b>	<b>8</b>
1. Tema și descrierea modelului real .....	8
2. Utilitatea acestuia.....	9
3. Funcționalitatea .....	9
<b>2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului....</b>	<b>10</b>
1. Tabela CASA_DISCURI.....	10
2. Tabela CONTRACT.....	10
3. Tabela MUZICIAN.....	11
4. Tabela TRUPA.....	11
5. Tabela MEMBRU_TRUPA.....	11
6. Tabela INSTRUMENT.....	12
7. Tabela CANTEC .....	12
8. Tabela TRUPA_CANTEC.....	12
9. Tabela ALBUM.....	13
10. Tabela CANTEC_ALBUM.....	13
11. Tabela ANGAJAT .....	14
12. Tabela MANAGER_TRUPA.....	14

13.	Tabela ADMINISTRATOR_CONTRACTE .....	15
14.	Tabela CERCETATOR_MARKETING.....	15
15.	Tabela INGINER_AUDIO .....	15
<b>3.</b>	<b>Descrierea entităților, incluzând precizarea cheii primare.....</b>	<b>16</b>
1.	Entitatea CASA_DISCURI.....	16
2.	Entitatea CONTRACT .....	16
3.	Entitatea MUZICIAN .....	16
4.	Entitatea MEMBRU_TRUPA .....	16
5.	Entitatea TRUPA.....	17
6.	Entitatea INSTRUMENT.....	17
7.	Entitatea TRUPA_CANTEC .....	17
8.	Entitatea CANTEC .....	17
9.	Entitatea CANTEC_ALBUM .....	17
10.	Entitatea ALBUM.....	18
11.	Entitatea ANGAJAT .....	18
12.	Entitatea MANAGER_TRUPA .....	18
13.	Entitatea ADMINISTRATOR_CONTRACTE.....	19
14.	Entitatea CERCETATOR_MARKETING .....	19
15.	Entitatea INGINER_AUDIO.....	19
<b>4.</b>	<b>Descrierea relațiilor, incluzând precizarea cardinalității acestora.....</b>	<b>20</b>

<b>5. Descrierea atributelor, inclusiv tipul de date și eventualele constrângeri, valori implicate, valori posibile ale atributelor.....</b>	23
1. Tabela CASA_DISCURI.....	23
2. Tabela CONTRACT.....	23
3. Tabela MUZICIAN.....	24
4. Tabela TRUPA.....	24
5. Tabela MEMBRU_TRUPA.....	25
6. Tabela INSTRUMENT.....	25
7. Tabela CANTEC .....	26
8. Tabela TRUPA_CANTEC.....	26
9. Tabela ALBUM.....	27
10. Tabela CANTEC_ALBUM.....	28
11. Tabela ANGAJAT.....	28
12. Tabela MANAGER_TRUPA.....	29
13. Tabela ADMINISTRATOR_CONTRACTE .....	29
14. Tabela CERCETATOR_MARKETING.....	30
15. Tabela INGINER_AUDIO .....	30
<b>6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.....</b>	31
<b>7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6 .....</b>	32

<b>8. Enumerarea schemelor relaţionale corespunzătoare diagramei conceptuale proiectate la punctul 7 .....</b>	33
<b>9. Realizarea normalizării până la forma normală 3 (FN1-FN3).....</b>	34
➤ FN1 – EXEMPLU DE NON-FN1.....	34
➤ FN2 – EXEMPLU DE NON-FN2.....	35
➤ FN3 – EXEMPLU DE NON-FN3.....	36
<b>10. Crearea unei secvenţe ce va fi utilizată în inserarea înregistrărilor în tabele.....</b>	38
<b>11. Crearea tabelelor în SQL şi inserarea de date coerente în fiecare dintre acestea .....</b>	40
1. Tabela CASA_DISCURI.....	40
2. Tabela MUZICIAN.....	42
3. Tabela CONTRACT.....	45
4. Tabela TRUPA.....	48
5. Tabela MEMBRU_TRUPA.....	50
6. Tabela INSTRUMENT.....	53
7. Tabela CANTEC.....	56
8. Tabela TRUPA_CANTEC.....	61
9. Tabela ALBUM.....	66
10. Tabela CANTEC_ALBUM.....	68
11. Tabela ANGAJAT.....	73

12.	Tabela MANAGER_TRUPA .....	78
13.	Tabela ADMINISTRATOR_CONTRACTE .....	80
14.	Tabela CERCETATOR_MARKETING.....	82
15.	Tabela INGINER_AUDIO .....	84
<b>12.</b>	<b>Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:.....</b>	<b>86</b>
➤	Cererea 1 .....	86
➤	Cererea 2 .....	88
➤	Cererea 3 .....	90
➤	Cererea 4 .....	93
➤	Cererea 5 .....	95
<b>13.</b>	<b>Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri .....</b>	<b>97</b>
➤	Operația 1.....	97
➤	Operația 2.....	98
➤	Operația 3.....	98
<b>14.</b>	<b>Crearea unei vizualizări complexe .....</b>	<b>100</b>
➤	operatie LMD permisa .....	100
➤	operatie LMD nepermisa.....	100
<b>15.</b>	<b>Formulați în limbaj natural și implementați în SQL cereri cu Outer Join, Division și Analiză-n.....</b>	<b>102</b>

➤ Cerere cu Outer Join.....	102
➤ Cerere cu Division.....	104
➤ Cerere cu analiză top-n.....	105
<b>16. Optimizarea unei cereri.....</b>	<b>106</b>
<b>17. Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării.....</b>	<b>110</b>
a. Realizarea normalizării BCNF, FN4, FN5. ....	110
b. Aplicarea denormalizării, justificând necesitatea acesteia. ....	112

---

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

---

### 1. Tema și descrierea modelului real

Tema acestui proiect este o bază de date care are scopul de a reține informații despre și de a gestiona una sau mai multe case de discuri.

Entitatea CASA\_DISCURI este entitatea care ține cont de informațiile unei case de discuri și care stochează informații importante despre aceasta. O casă de discuri poate să aibă contracte cu muzicieni și poate să dețină albume. Albumele sunt aparținute unei singure trupe și pot fi formate din unul sau mai multe cântece. Fiecare cântec poate să aibă unul sau mai mulți artiști și poate să apară în unul sau mai multe albume (precum albume de „Best Hits”). În acest proiect cuvântul trupă nu este folosit în modul convențional, referindu-se mai mult la numele unei entități care lansează cântece și albume. O trupă poate să conțină un singur muzician sau mai mulți. Un muzician poate să aparțină unei sau mai multor trupe. Am realizat acest lucru pentru a diferenția între muzician ca persoană și trupă ca entitate care scoate cântece și albume, deoarece pot exista artiști solo care scot albume sau realizează muzică sub nume diferite, aliasuri sau care pur și simplu aparțin mai multor trupe. Acest lucru de asemenea evită legarea muzicienilor, însă și a trupelor de atât album și cântec - în modul acesta doar trupa este legată de cântec și album. Fiecare trupă poate să aibă mai multe instrumente. Casa de discuri are, de asemenea, cel puțin un angajat, care are posibilitatea de a fi manager de trupă, care este asociat unei singure trupe, administrator de contracte, cercetator de marketing sau inginer audio.

## 2. Utilitatea acestuia

Această bază de date poate fi utilizată pentru a ține cont de și a administra angajați, trupe, albume, contracte sau alte entități aferente uneia sau mai multor case de discuri.

## 3. Funcționalitatea

Funcționalitatea această a bazei de date include stocarea și gestionarea informațiilor despre casa de discuri însă și despre contractele, trupele, muzicienii, instrumentele, albumele, cântecele și angajații aferente acelei case de discuri. Aceasta permite actualizarea, citirea, crearea și ștergerea informațiilor de care este nevoie în a gestiona o casă de discuri. Această bază de date ar fi utilă unei companii de case de discuri care dorește să își stocheze informațiile despre atributile sale într-un mod eficient, util și optim, sau chiar și unei companii mamă, precum Warner Records, care dețin mai multe case de discuri.

---

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

---

### 1. Tabela CASA\_DISCURI

- ID\_CASA\_DISCURI este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- NUME trebuie să aibă maxim 40 de caractere și să fie nenul și unic.
- ADRESA trebuie să aibă maxim 255 de caractere.
- WEBSITE trebuie să aibă maxim 255 de caractere.
- AN\_INFINTARE trebuie să fie o valoarea întreagă, mai mare sau egală cu 1889 și mai mică sau egală decât anul curent (2024).

### 2. Tabela CONTRACT

- ID\_CONTRACT este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_CASA\_DISCURI este o cheie externă către tabela CASA\_DISCURI și trebuie să se refere la un ID\_CASA\_DISCURI existent în acel tabel.
- ID\_MUZICIAN este o cheie externă către tabela MUZICIAN și trebuie să se refere la un ID\_MUZICIAN existent în acel tabel.
- DATA\_SEMNARE și DATA\_EXPIRARE sunt valori de tip dată și sunt nenele.

### 3. Tabela MUZICIAN

- ID\_MUZICIAN este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- NUME trebuie să aibă maxim 40 de caractere și să fie nenul.
- ZI\_NASTERE este o valoare de tip dată și este nenulă.
- GEN trebuie să aibă maxim 20 de caractere.

### 4. Tabela TRUPA

- ID\_TRUPA este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- NUME trebuie să aibă maxim 40 de caractere și să fie nenul și unic.
- GEN trebuie să aibă maxim 60 de caractere și să fie nenul.
- DATA\_INFIINTARE este o valoare de tip dată și este nenulă.

### 5. Tabela MEMBRU\_ TRUPA

- ID\_MEMBRU\_TRUPA este cheia primară formată din ID\_MUZICIAN și ID\_TRUPA și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_MUZICIAN este o cheie externă către tabela MUZICIAN și trebuie să se refere la un ID\_MUZICIAN existent în acel tabel.
- ID\_TRUPA este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- POZITIE trebuie să aibă maxim 40 de caractere, să fie nenulă și să facă parte din mulțimea: Vocalist, Chitarist, Basist, Claviatura, Tobosar, Pianist.

## 6. Tabela INSTRUMENT

- ID\_INSTRUMENT este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_TRUPA este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- NUME trebuie să aibă maxim 40 de caractere și să fie nenul.
- UTILIZATOR trebuie să aibă maxim 40 de caractere și să facă parte din mulțimea: Vocalist, Chitarist, Basist, Claviatura, Tobosar, Pianist.

## 7. Tabela CANTEC

- ID\_CANTEC este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- NUME trebuie să aibă maxim 255 de caractere și să fie nenul.
- LUNGIME trebuie să fie un număr de maxim 4 cifre și maxim 2 zecimale, nenul și pozitiv.

## 8. Tabela TRUPA\_CANTEC

- ID\_TRUPA\_CANTEC este cheia primară formată din ID\_TRUPA și ID\_CANTEC și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_TRUPA este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- ID\_CANTEC este o cheie externă către tabela CANTEC și trebuie să se refere la un ID\_CANTEC existent în acel tabel.

## 9. Tabela ALBUM

- ID\_ALBUM este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_TRUPA este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- ID\_CASA\_DISCURI este o cheie externă către tabela CASA\_DISCURI și trebuie să se refere la un ID\_CASA\_DISCURI existent în acel tabel.
- NUME trebuie să aibă maxim 255 de caractere și să fie nenul.
- TIP trebuie să aibă maxim 10 de caractere, să fie nenul și să facă parte din mulțimea: EP, Album, Single.
- DATA\_LANSARE să fie de tip dată.

## 10. Tabela CANTEC\_ALBUM

- ID\_CANTEC\_ALBUM este cheia primară formată din ID\_CANTEC și ID\_ALBUM și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_CANTEC este o cheie externă către tabela CANTEC și trebuie să se refere la un ID\_CANTEC existent în acel tabel.
- ID\_ALBUM este o cheie externă către tabela ALBUM și trebuie să se refere la un ID\_ALBUM existent în acel tabel.

## 11. Tabela ANGAJAT

- ID\_ANGAJAT este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_CASA\_DISCURI este o cheie externă către tabela CASA\_DISCURI și trebuie să se refere la un ID\_CASA\_DISCURI existent în acel tabel.
- NUME trebuie să aibă maxim 40 de caractere și să fie nenul.
- SALARIU trebuie să fie un număr nenul, pozitiv.
- DEPARTAMENT trebuie să aibă maxim 40 de caractere și să fie nenul.
- DATA\_ANGAJARE să fie de tip dată, nenulă.

## 12. Tabela MANAGER\_TRUPA

- ID\_MANAGER\_TRUPA este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_TRUPA este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- ID\_ANGAJAT este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel.
- O trupă poate să aibă manager doar dacă toți membrii acelei trupe au contract cu respectiva casă de discuri.

### 13. Tabela ADMINISTRATOR\_CONTRACTE

- ID\_ADMINISTRATOR\_CONTRACTE este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_ANGAJAT este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel
- SPECIALITATE trebuie să aibă maxim 40 de caractere.

### 14. Tabela CERCETATOR\_MARKETING

- ID\_CERCETATOR\_MARKETING este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_ANGAJAT este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel.
- CATEGORIE trebuie să aibă maxim 40 de caractere.

### 15. Tabela INGINER\_AUDIO

- ID\_INGINER\_AUDIO este cheia primară și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_ANGAJAT este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel.
- SPECIALITATE\_GEN trebuie să aibă maxim 40 de caractere.

---

---

### 3. Descrierea entităților, incluzând precizarea cheii primare.

---

---

## 1. Entitatea CASA\_DISCURI

Entitatea CASA\_DISCURI conține informații despre casele de discuri, având cheia primară ID\_CASA\_DISCURI și conține informații despre fiecare casă, precum numele, adresa website, și anul în care a fost înființată.

## 2. Entitatea CONTRACT

Entitatea CONTRACT conține informații despre contractele încheiate între o casă de discuri și un muzician, având cheia primară ID\_CONTRACT și deținând alte informații precum dată semnare a contractului și data de expirare a acestuia. Aceasta are de asemenea 2 chei externe ID\_MUZICIAN și ID\_CASA\_DISCURI care leagă câte un muzician de către o casă de discuri printr-un contract.

## 3. Entitatea MUZICIAN

Entitatea MUZICIAN conține informații despre fiecare muzician în parte, având cheia primară ID\_MUZICIAN și stochează și alte informații precum nume, ziua de naștere și genul acestuia.

## 4. Entitatea MEMBRU\_TRUPA

Entitatea MEMBRU\_TRUPA este un tabel asociativ pentru o relație de tip many to many între tabelele MUZICIAN și TRUPA. Această entitate este utilizată pentru a corespunde ce muzicieni aparțin căror trupe, folosind logica că muzicienii pot aparține mai multor trupe iar fiecare trupă are mai mulți muzicieni. Această entitate are cheia primară ID\_MEMBRU\_TRUPA, formată din cheile străine ID\_MUZICIAN și ID\_TRUPA, și mai are

de asemenea un atribut care reține poziția muzicianului în trupă (cum ar fi ghitarist sau vocalist).

## 5. Entitatea TRUPA

Entitatea TRUPA stochează informații despre fiecare trupă, precum nume, gen și data de înființare a acestea. Are cheia primară ID\_TRUPA.

## 6. Entitatea INSTRUMENT

Entitatea instrument reține informații despre fiecare instrument, numele acestuia, ce fel de muzician poate utiliza acel instrument, ID\_INSTRUMENT care este cheia primară și are o cheie străină ID\_TRUPA care asociază fiecare instrument cu trupa căreia aparține

## 7. Entitatea TRUPA\_CANTEC

Entitatea TRUPA\_CANTEC este o tabelă asociativă pentru o relație de tip many to many între tabelele TRUPA și CANTEC. Această entitate este utilizată pentru a corela atât toate cântecele cântate de un muzician însă și toți muzicienii care cântă un cântec (un exemplu ar fi Telephone, cântat de Lady Gaga însă și de Beyoncé). Are cheia primară ID\_TRUPA\_CANTEC care este formată din cheile străine ID\_TRUPA și ID\_CANTEC.

## 8. Entitatea CANTEC

Entitatea CANTEC stochează informații despre fiecare cântec și are attributele nume și lungimea în minute, alături de cheia primară ID\_CANTEC.

## 9. Entitatea CANTEC\_ALBUM

Entitatea CANTEC\_ALBUM este o tabelă asociativă pentru o relație de tip many to many între tabelele ALBUM și CANTEC. Această entitate este utilizată pentru a corela

atât toate cântecele care apar într-un album însă și toate albumele în care poate să apară un cântec. Logica din spatele acestui tabel este faptul că pot exista albume de tip ‚single’, ‚EP’ sau ‚album de lungime întreagă’ precum și albume ‚best hits’ sau compilații de cântece lansate tot ca albume în care cântecele se pot repeta (un exemplu comun ar fi un cântec care este lansat ca single înainte de lansarea integrală a albumului iar acesta poate fi găsit și ca single și ca parte din albumul final). Tabela are cheia primară ID\_CANTEC\_ALBUM, formată din cheile străine ID\_ALBUM și ID\_CANTEC.

## 10. Entitatea ALBUM

Entitatea ALBUM stochează informații despre fiecare album în parte, precum nume, data lansării și tip, însă are de asemenea și 2 chei străine: ID\_TRUPA fiind id-ul trupei sub cărei nume albumul este lansat și ID\_CASA\_DISCURI fiind id-ul casei de discuri care deține drepturile de autor asupra albumului. Are cheia primară ID\_ALBUM.

## 11. Entitatea ANGAJAT

Entitatea ANGAJAT reține informații despre fiecare angajat al unei case de discuri, având cheia primară ID\_ANGAJAT și atrbute precum nume, salariu, departament și data angajării. Are de asemenea o cheie străină ID\_CASA\_DISCURI care indică în ce casă de discuri lucrează acel angajat.

## 12. Entitatea MANAGER\_TRUPA

Entitatea MANAGER\_TRUPA este o sub entitate a entității ANGAJAT în care fiecare manager de trupă este obligatoriu un angajat la rândul lui. Această tabelă are

cheia primară ID\_MANAGER\_TRUPA și 2 chei secundare: ID\_TRUPA, reprezentând trupa de care acest manager este responsabil, și ID\_ANGAJAT, pentru a face legătura cu tabela ANGAJAT. O trupă are un manager de trupă asociat acelei case de discuri dacă și numai dacă au sau au avut un contract cu acea casa de discuri.

### 13. Entitatea ADMINISTRATOR\_CONTRACTE

Entitatea ADMINISTRATOR\_CONTRACTE este o sub entitate a entității ANGAJAT în care fiecare administrator de contracte este obligatoriu angajat la rândul lui. Aceasta are cheia primară ID\_ADMINISTRATOR\_CONTRACTE și are un atribut care menționează specializarea, alături de cheia străină ID\_ANGAJAT care face legătura cu tabela ANGAJAT.

### 14. Entitatea CERCETATOR\_MARKETING

Entitatea CERCETATOR\_MARKETING este o sub entitate a entității ANGAJAT în care fiecare cercetător de marketing este obligatoriu angajat la rândul lui. Această tabelă are cheia primară ID\_CERCETATOR\_MARKETING și are un atribut care menționează categoria, alături de cheia străină ID\_ANGAJAT care face legătura cu tabela ANGAJAT.

### 15. Entitatea INGINER\_AUDIO

Entitatea INGINER\_AUDIO este o sub entitate a entității ANGAJAT în care fiecare inginer audio este obligatoriu angajat la rândul lui. Această tabelă are cheia primară ID\_INGINER\_AUDIO și are un atribut care menționează specialitatea în genul de muzică a fiecărui inginer, alături de cheia străină ID\_ANGAJAT care face legătura cu tabela ANGAJAT.

---

**4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

---

- Tabela CASA\_DISCURI are o relație de tip „one to many” cu tabela CONTRACT, deoarece o casă de discuri poate să aibă zero sau mai multe contracte însă în clauza unui contract există obligatoriu o singură casă de discuri. Astfel, cardinalitatea este 1(1):M(0).
- Tabela ANGAJAT are o relație de tip „many to one” cu tabela CASA\_DISCURI deoarece un angajat aparține obligatoriu unei singure case de discuri iar fiecare casă de discuri are unul sau mai mulți angajați. Astfel cardinalitatea este M(1):1(1).
- Tabelele MANAGER\_TRUPA, ADMINISTRATOR\_CONTRACTE, CERCETATOR\_MARKETING, INGINER\_AUDIO au toate câte o relație de tip „one to one” cu tabela ANGAJAT, deoarece fiecare entitate din cele menționate anterior sunt un tip de angajat. Astfel cardinalitatea este 1(0):1.
- Tabela ALBUM are o relație de tip „many to one” cu tabela CASA\_DISCURI, deoarece fiecare album este deținut de o casă de discuri însă o casă de discuri poate să aibă mai multe albume. Astfel cardinalitatea este M(0):1(1).
- Tabela CANTEC are o relație de tip „many to many” cu tabela ALBUM, deoarece un cântec poate să apară pe mai multe albume și un album are mai multe cântece. Pentru a face corespondență între aceste 2 tabele într-un mod corect există tabela CANTEC\_ALBUM care are relație de tip „many to one”, de cardinalitate M(1):1(1),

atât cu tabela CANTEC, dar și cu tabela ALBUM. Cardinalitatea dintre CANTEC și ALBUM ar fi M(1):M(1).

- Tabela CANTEC are o relație de tip „many to many” cu tabela TRUPA, deoarece un cântec poate fi când dat de mai mulți artiști și fiecare artist poate cânta mai multe cântece. Pentru a face corespondență între aceste 2 tabele într-un mod corect există tabela TRUPA\_CANTEC care are relație de tip „many to one”, de cardinalitate M(1):1(1) cu tabela CANTEC, respectiv M(0):1(1) cu tabela TRUPA. Cardinalitatea dintre CANTEC și TRUPA ar fi M(0):M(1).
- Tabela MUZICIAN are o relație de tip „one to one” cu tabela CONTRACT, deoarece într-un contract este întocmit cu câte un muzician. Astfel cardinalitatea este 1(1):1(0).
- Tabela TRUPA are o relație de tip „one to one” cu tabela MANAGER\_TRUPA, deoarece fiecare trupă poate să aiba un manager, dacă toți muzicienii din acea trupă au contract cu casa de discuri. Astfel, cardinalitatea este 1(1):1(0).
- Tabela TRUPA are o relație de tip „one to many” cu tabela ALBUM, deoarece fiecare trupă poate să scoată mai multe albume. Astfel, cardinalitatea este 1(1):M(0).
- Tabela INSTRUMENT are o relație de tip „many to one” cu tabela TRUPA, deoarece fiecare trupă poate să dețină mai multe instrumente. Astfel, cardinalitatea este M(0):1(1).

- Tabela MUZICIAN are o relație de tip „many to many” cu tabela TRUPA, deoarece un muzician poate apartine uneia sau mai multor trupe și o trupă poate fi compusă din unul sau mai mulți muzicieni. Pentru a face corespondența între aceste 2 tabele într-un mod corect există tabela MEMBRU\_TRUPA care are relație de tip „many to one”, de cardinalitate M(1):1(1) cu tabela MUZICIAN, respectiv tabela TRUPA. Cardinalitatea dintre MUZICIAN și TRUPA ar fi M(1):M(1).

---

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

---

## 1. Tabela CASA\_DISCURI

- ID\_CASA\_DISCURI (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- NUME(varchar2(40)): numele casei de discuri, valoare nenulă și unică.
- ADRESA(varchar2(255)): adresa casei de discuri .
- WEBSITE(varchar2(255)): adresa website-ului casei de discuri.
- AN\_INFIINTARE(number): anul înființării casei de discuri, mai mare sau egal cu 1889 și mai mic sau egal decât anul curent (2024).

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_CASA_DISCURI	NUMBER(38,0)	No	"LUCA"."CASA_DISCURI_SEQ"."NEXTVAL"	1	(null)
2 NUME	VARCHAR2(40 BYTE)	No	(null)	2	(null)
3 ADRESA	VARCHAR2(255 BYTE)	Yes	(null)	3	(null)
4 WEBSITE	VARCHAR2(255 BYTE)	Yes	(null)	4	(null)
5 AN_INFIINTARE	NUMBER	Yes	(null)	5	(null)

## 2. Tabela CONTRACT

- ID\_CONTRACT (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_CASA\_DISCURI (int, FK): este o cheie externă către tabela CASA\_DISCURI și trebuie să se refere la un ID\_CASA\_DISCURI existent în acel tabel.
- ID\_MUZICIAN (int, FK): este o cheie externă către tabela MUZICIAN și trebuie să se refere la un ID\_MUZICIAN existent în acel tabel.

- DATA\_SEMNARE și DATA\_EXPIRARE (date): nenule, reprezintă data de semnare a contractului respectiv data de expirare a acestuia.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_CONTRACT	NUMBER (38,0)	No	"LUCA"."CONTRACT_SEQ"."NEXTVAL"	1	(null)
2 ID_MUZICIAN	NUMBER (38,0)	Yes	(null)	2	(null)
3 ID_CASA_DISCURI	NUMBER (38,0)	Yes	(null)	3	(null)
4 DATA_SEMNARE	DATE	No	(null)	4	(null)
5 DATA_EXPIRARE	DATE	No	(null)	5	(null)

### 3. Tabela MUZICIAN

- ID\_MUZICIAN (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- NUME(varchar2(40)): nenul, reprezintă numele muzicianului.
- ZI\_NASTERE (date): reprezintă ziua de naștere a muzicianului.
- GEN(varchar2(20)): reprezintă genul muzicianului.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_MUZICIAN	NUMBER (38,0)	No	"LUCA"."MUZICIAN_SEQ"."NEXTVAL"	1	(null)
2 NUME	VARCHAR2 (40 BYTE)	No	(null)	2	(null)
3 ZI_NASTERE	DATE	Yes	(null)	3	(null)
4 GEN	VARCHAR2 (20 BYTE)	Yes	(null)	4	(null)

### 4. Tabela TRUPA

- ID\_TRUPA (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- NUME (varchar2(40)): nenul și unic, reprezintă numele trupei.
- GEN(varchar2(60)): nenul reprezintă genul muzical al trupei.
- DATA\_INFIINTARE (date): nenul reprezintă data de înființarea trupei.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_TRUPA	NUMBER(38,0)	No	"LUCA"."TRUPA_SEQ"."NEXTVAL"	1	(null)
2 NUME	VARCHAR2(40 BYTE)	No	(null)	2	(null)
3 GEN	VARCHAR2(60 BYTE)	No	(null)	3	(null)
4 DATA_INFIINTARE	DATE	No	(null)	4	(null)

## 5. Tabela MEMBRU\_ TRUPA

- ID\_MEMBRU\_TRUPA (int, PK): este cheia primară formată din ID\_MUSICIAN și ID\_TRUPA și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_MUSICIAN (int, FK): este o cheie externă către tabela MUSICIAN și trebuie să se refere la un ID\_MUSICIAN existent în acel tabel.
- ID\_TRUPA (int, FK): este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- POZITIE (varchar2(40)): poziția membrului în trupă respectiv pe scenă și trebuie să facă parte din mulțimea: Vocalist, Chitarist, Basist, Claviatura, Tobosar, Pianist.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_MUSICIAN	NUMBER(38,0)	No	(null)	1	(null)
2 ID_TRUPA	NUMBER(38,0)	No	(null)	2	(null)
3 POZITIE	VARCHAR2(40 BYTE)	No	(null)	3	(null)

## 6. Tabela INSTRUMENT

- ID\_INSTRUMENT (int, PK): cheia primară a tabelului, cu valoarea default data de secvență.
- ID\_TRUPA (int, FK): este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.

- NUME (varchar2(40)): nenul, numele instrumentului.
- UTILIZATOR(varchar2(40)): reprezintă tipul de muzician care ar trebui să folosească acest instrument și trebuie să facă parte din mulțimea: Vocalist, Chitarist, Basist, Claviatura, Tobosar, Pianist.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_INSTRUMENT	NUMBER (38, 0)	No	"LUCA"."INSTRUMENT_SEQ"."NEXTVAL"	1	(null)
2 ID_TRUPA	NUMBER (38, 0)	Yes	(null)	2	(null)
3 NUME	VARCHAR2 (40 BYTE)	No	(null)	3	(null)
4 UTILIZATOR	VARCHAR2 (40 BYTE)	Yes	(null)	4	(null)

## 7. Tabela CANTEC

- ID\_CANTEC (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- NUME(varchar2(40)): nenul, reprezintă numele cântecului.
- LUNGIME (number(4,2)): reprezintă lungimea cântecului în minute și trebuie să fie nenul și strict pozitiv.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_CANTEC	NUMBER (38, 0)	No	"LUCA"."CANTEC_SEQ"."NEXTVAL"	1	(null)
2 NUME	VARCHAR2 (255 BYTE)	No	(null)	2	(null)
3 LUNGIME	NUMBER (4, 2)	No	(null)	3	(null)

## 8. Tabela TRUPA\_CANTEC

- ID\_TRUPA\_CANTEC (int, PK): este cheia primară formată din ID\_TRUPA și ID\_CANTEC și trebuie să fie unică pentru fiecare înregistrare din tabel.

- ID\_TRUPA (int, FK): este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- ID\_CANTEC (int, FK): este o cheie externă către tabela CANTEC și trebuie să se refere la un ID\_CANTEC existent în acel tabel.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_TRUPA	NUMBER (38, 0)	No	(null)	1	(null)
2 ID_CANTEC	NUMBER (38, 0)	No	(null)	2	(null)

## 9. Tabela ALBUM

- ID\_ALBUM (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_TRUPA (int, FK): este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- ID\_CASA\_DISCURI (int, FK): este o cheie externă către tabela CASA\_DISCURI și trebuie să se refere la un ID\_CASA\_DISCURI existent în acel tabel.
- NUME (varchar2(225)): nenul reprezintă titlul albumului.
- TIP (varchar2(10)): reprezintă tipul de album și trebuie să facă parte din mulțimea: EP, Album, Single.
- DATA\_LANSARE (date): reprezintă data de lansare a albumului.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ALBUM	NUMBER (38, 0)	No	"LUCA"."ALBUM_SEQ"."NEXTVAL"	1	(null)
2 ID_TRUPA	NUMBER (38, 0)	Yes	(null)	2	(null)
3 ID_CASA_DISCURI	NUMBER (38, 0)	Yes	(null)	3	(null)
4 NUME	VARCHAR2 (255 BYTE)	No	(null)	4	(null)
5 TIP	VARCHAR2 (10 BYTE)	No	(null)	5	(null)
6 DATA_LANSARE	DATE	Yes	(null)	6	(null)

## 10. Tabela CANTEC\_ALBUM

- ID\_CANTEC\_ALBUM (int, PK): este cheia primară formată din ID\_CANTEC și ID\_ALBUM și trebuie să fie unică pentru fiecare înregistrare din tabel.
- ID\_CANTEC (int, FK): este o cheie externă către tabela CANTEC și trebuie să se refere la un ID\_CANTEC (int, FK): existent în acel tabel.
- ID\_ALBUM (int, FK): este o cheie externă către tabela ALBUM și trebuie să se refere la un ID\_ALBUM existent în acel tabel.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ALBUM	NUMBER (38, 0)	No	(null)	1 (null)	
2 ID_CANTEC	NUMBER (38, 0)	No	(null)	2 (null)	

## 11. Tabela ANGAJAT

- ID\_ANGAJAT (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_CASA\_DISCURI (int, FK): este o cheie externă către tabela CASA\_DISCURI și trebuie să se refere la un ID\_CASA\_DISCURI existent în acel tabel.
- NUME (varchar2(40)): nenul reprezintă numele angajatului.
- SALARIU (number): nenul și pozitiv, reprezintă salariul angajatului.
- DEPARTAMENT(varchar2(40)): nenul, reprezintă numele departamentului la care angajatul lucrează.
- DATA\_ANGAJARE (date): nenul, reprezintă data la care angajatul a fost angajat.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ANGAJAT	NUMBER(38,0)	No	"LUCA"."ANGAJAT_SEQ"."NEXTVAL"	1	(null)
2 ID_CASA_DISCURI	NUMBER(38,0)	Yes	(null)	2	(null)
3 NUME	VARCHAR2(40 BYTE)	No	(null)	3	(null)
4 SALARIU	NUMBER	No	(null)	4	(null)
5 DEPARTAMENT	VARCHAR2(40 BYTE)	No	(null)	5	(null)
6 DATA_ANGAJARE	DATE	No	(null)	6	(null)

## 12.Tabela MANAGER\_TRUPA

- ID\_MANAGER\_TRUPA (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_TRUPA (int, FK): este o cheie externă către tabela TRUPA și trebuie să se refere la un ID\_TRUPA existent în acel tabel.
- ID\_ANGAJAT (int, FK): este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_MANAGER_TRUPA	NUMBER(38,0)	No	"LUCA"."MANAGER_TRUPA_SEQ"."NEXTVAL"	1	(null)
2 ID_TRUPA	NUMBER(38,0)	Yes	(null)	2	(null)
3 ID_ANGAJAT	NUMBER(38,0)	Yes	(null)	3	(null)

## 13.Tabela ADMINISTRATOR\_CONTRACTE

- ID\_ADMINISTRATOR\_CONTRACTE (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_ANGAJAT (int, FK): este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel
- SPECIALITATE (varchar2(40)): reprezintă domeniul de specialitate al administratorului de contracte.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ADMINISTRATOR_CONTRACTE	NUMBER(38,0)	No	"LUCA"."ADMINISTRATOR_CONTRACTE_SEQ"."NEXTVAL"	1	(null)
2 ID_ANGAJAT	NUMBER(38,0)	Yes	(null)	2	(null)
3 SPECIALITATE	VARCHAR2(40 BYTE)	Yes	(null)	3	(null)

## 14. Tabela CERCETATOR\_MARKETING

- ID\_CERCETATOR\_MARKETING (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_ANGAJAT (int, FK): este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel.
- CATEGORIE (varchar2(40)): reprezintă categoria din care face parte respectivul cercetător de marketing.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_CERCETATOR_MARKETING	NUMBER (38, 0)	No	"LUCA"."CERCETATOR_MARKETING_SEQ"."NEXTVAL"	1	(null)
2 ID_ANGAJAT	NUMBER (38, 0)	Yes	(null)	2	(null)
3 CATEGORIE	VARCHAR2 (40 BYTE)	Yes	(null)	3	(null)

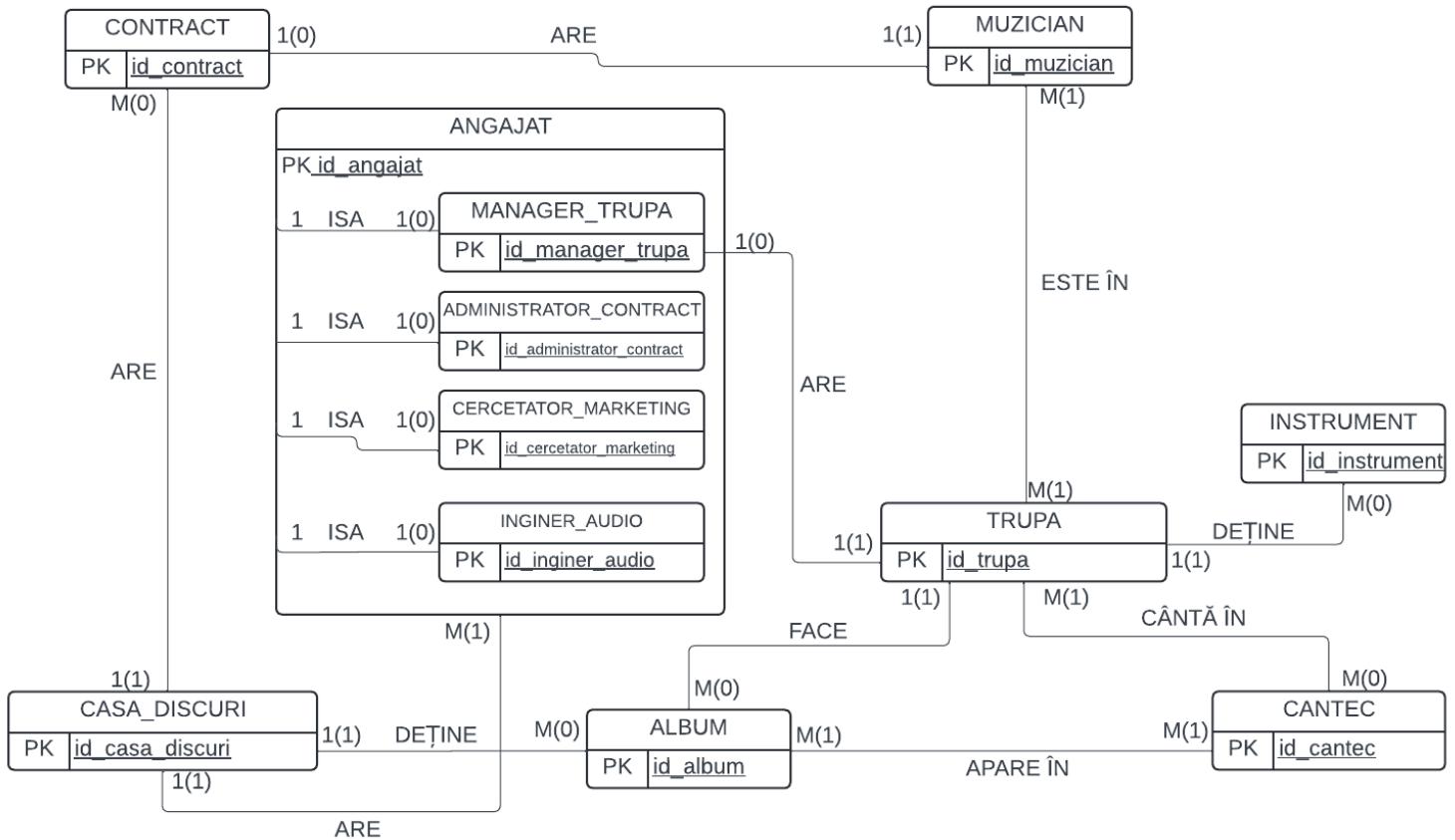
## 15. Tabela INGINER\_AUDIO

- ID\_INGINER\_AUDIO (int, PK): cheia primară a tabelului, cu valoarea default dată de secvență.
- ID\_ANGAJAT (int, FK): este o cheie externă către tabela ANGAJAT și trebuie să se refere la un ID\_ANGAJAT existent în acel tabel.
- SPECIALITATE\_GEN (varchar2(40)): reprezintă numele genului de muzică în care respectivul inginer audio se specializează.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_INGINER_AUDIO	NUMBER (38, 0)	No	"LUCA"."INGINER_AUDIO_SEQ"."NEXTVAL"	1	(null)
2 ID_ANGAJAT	NUMBER (38, 0)	Yes	(null)	2	(null)
3 SPECIALITATE_GEN	VARCHAR2 (40 BYTE)	Yes	(null)	3	(null)

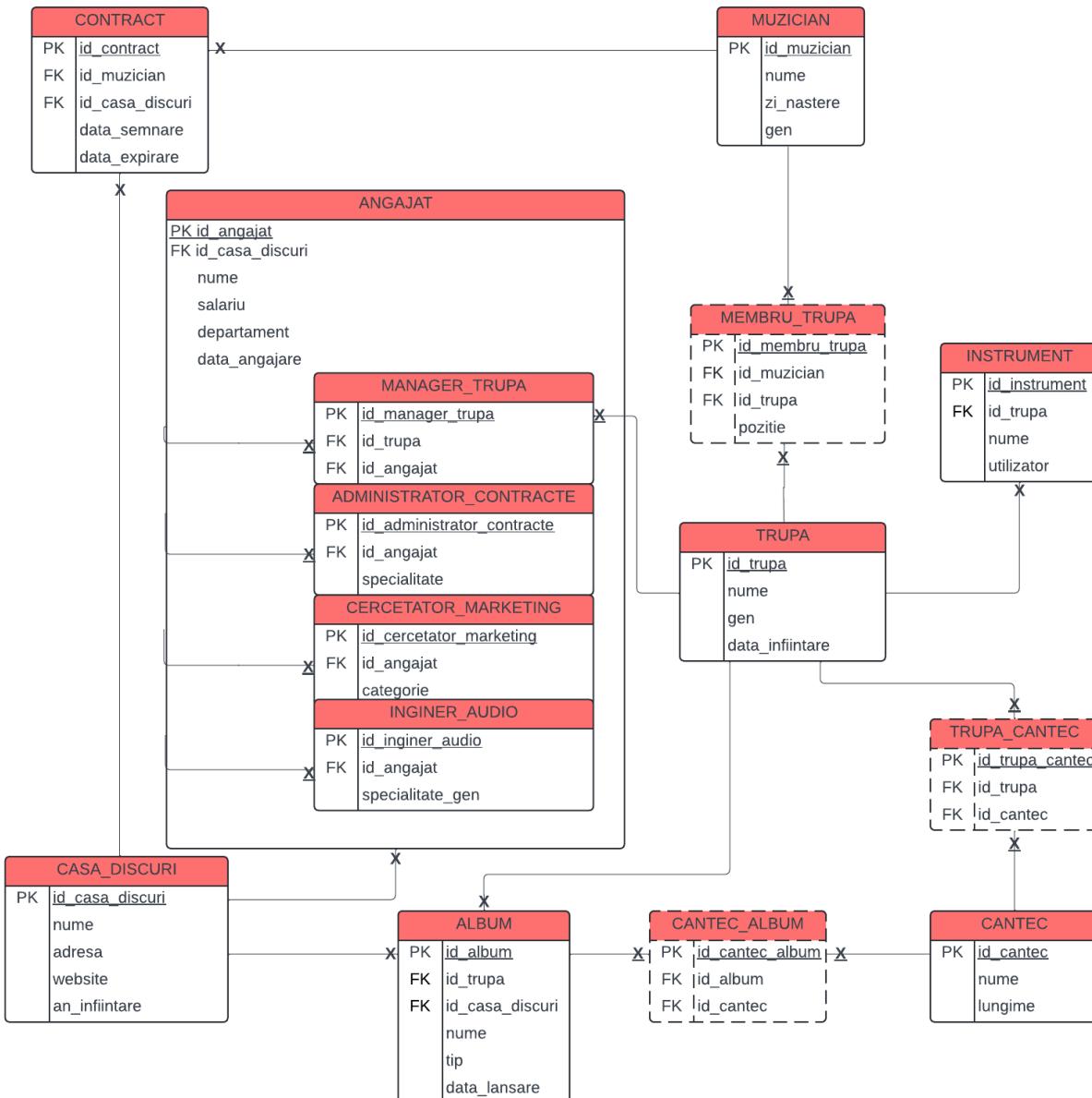
**6. Realizarea diagramei entitate-relaţie corespunzătoare descrierii de la punctele**

**3-5.**



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

Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



---

**8. Enumerarea schemelor relationale corespunzătoare diagramei conceptuale  
proiectate la punctul 7.**

---

- CASA\_DISCURI (id\_casa\_discuri, nume, adresa, website, an\_infiintare)
- CONTRACT (id\_contract, id\_muzician, id\_casa\_discuri, data\_semnare, data\_expirare)
- MUZICIAN (id\_muzician, nume, zi\_nastere, gen)
- MEMBRU\_TRUPA (id\_membru\_trupa, id\_muzician, id\_trupa, pozitie)
- TRUPA (id\_trupa, nume, gen, data\_infiintare)
- INSTRUMENT (id\_instrument, id\_trupa, nume, utilizator)
- TRUPA\_CANTEC (id\_trupa\_cantec, id\_cantec, id\_trupa)
- CANTEC (id\_cantec, nume, lungime)
- CANTEC\_ALBUM (id\_cantec\_album, id\_album, id\_cantec)
- ALBUM (id\_album, id\_trupa, id\_casa\_discuri, nume, tip, data\_lansarii)
- ANGAJAT (id\_angajat, id\_casa\_discuri, nume, salariu, departament, data\_angajare)
- MANAGER\_TRUPA (id\_manager\_trupa, id\_trupa, id\_angajat)
- ADMINISTRATOR\_CONTRACTE (id\_administrator\_contracte, id\_angajat, specialitate)
- CERCETATOR\_MARKETING (id\_cercetator\_marketing, id\_angajat, categorie)
- INGINER\_AUDIO (id\_ingenier\_audio, id\_angajat, specialitate\_gen)

---

---

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

---

---

➤ FN1 – EXEMPLU DE NON-FN1

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare care nu se poate descompune.

Pentru a da un exemplu NON-FN1 voi utiliza tabelele MUZICIAN și TRUPA. În cadrul unei trupe, membrii acesteia nu se repeta, dar la nivelul întregului tabel, un muzician poate să aparțină mai multor trupe. Ca să rezolvăm această situație, construim un tabel nou MEMBRU\_TRUPA care face legătura dintre muzicieni și trupele cărora aparțin, transformarea în FN1 arătând astfel:

ID_TRUPA	ID_MUZICIAN
1	1,2,4
3	1,3
9	2,4,7



ID_TRUPA	ID_MUZICIAN
1	1
1	2
1	4
3	1
3	3
9	2
9	4
9	7

Cheia primară a acestei tabele noi construite, MEMBRU\_TRUPA, care face corespondența între muzician și o trupa în care acesta apare, va fi compusă din id-ul unei trupe și id-ul unui muzician.

## ➤ FN2 – EXEMPLU DE NON-FN2

O relație se află în FN2 dacă și numai dacă această relație este deja în FN1, iar fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară.

Pentru a exemplifica acest lucru în baza mea de date voi folosi tabela TRUPA\_CANTEC. Aceasta se află deja în FN1 deoarece are câte un identificator unic pentru toate cheile. Pentru a fi în FN2, fiecare atribut care nu este cheie primară sau nu participă la cheia primară trebuie să fie dependent de întreaga cheie primară. În tabela noastră avem ca și cheie primară ID\_TRUPA și ID\_CANTEC. Putem observa faptul că NUME\_TRUPA nu este cheie primară și nu depinde de întreaga cheie primară compusă de cheile primare menționate anterior, observându-se mai degrabă o dependență directă între NUME\_TRUPA și ID\_TRUPA. În mod similar putem observa o dependență directă între NUME\_CANTEC și ID\_CANTEC. Așadar, tabela noastră nu se află momentan în FN2. Pentru a remedia situația, observăm că atrbutele NUME\_TRUPA și NUME\_CANTEC trebuie să se afle doar în tabelele trupă respectiv cântec. Astfel, transformarea în FN2 va arata în modul următor:

Tabela TRUPA\_CANTEC

ID_TRUPA	NUME_TRUPA	ID_CANTEC	NUME_CANTEC
1	”My Chemical Romance”	13	”The End.”
1	”My Chemical Romance”	14	”Vampire Money”
2	”TWRP”	44	”Digital Nightmare”
2	”TWRP”	48	”Online”
7	”Tom Cardy”	48	”Online”

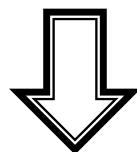


Tabela nouă TRUPA\_CANTEC

ID_TRUPA	ID_CANTEC
1	13
1	14
2	44
2	48
7	48

Iar atributele NUME\_TRUPA și NUME\_CANTEC au fost mutate în tabelele TRUPA, respectiv CANTEC, corespunzătoare cheii primare potrivite.

➤ FN3 – EXEMPLU DE NON-FN3

O relație se află în FN3 dacă și numai dacă această relație este deja în FN2 (și implicit în FN1), iar fiecare atribut care nu este cheie primară este dependent de cheia primară.

Voi folosi tabela CONTRACT pentru a exemplifica această proprietate. Inițial, aceasta conținea ca attribute ID\_CONTRACT, ID\_MUSICIAN, NUME\_CASA\_DISCURI, ADRESA\_CASA\_DISCURI, WEBSITE\_CASA\_DISCURI, AN\_INFIINTARE\_CASA\_DISCURI DATA\_SEMNARE\_CONTRACT și DATA\_EXPIRARE\_CONTRACT. Putem observa că se află atât în FN1 cât și în FN2. Atributele DATA\_SEMNARE\_CONTRACT și DATA\_EXPIRARE\_CONTRACT depindează în mod direct de ID\_CONTRACT, care este cheia primară. Putem observa însă că attributele ADRESA\_CASA\_DISCURI, WEBSITE\_CASA\_DISCURI, AN\_INFIINTARE\_CASA\_DISCURI depind în mod direct de NUME\_CASA\_DISCURI.

Pentru a rezolva acest caz și a aduce tabela în relație FN3 vom construi un tabel nou numit CASA\_DISCURI și vom lua atributele dependente de NUME\_CASA\_DISCURI din CONTRACT și le ducem în tabela nouă. În tabela CONTRACT voi înlocui toate atributele scoase cu o cheie străină numită ID\_CASA\_DISCURI, care face corespondență între contractul și casa de discuri aferentă. Va apărea o relație de tip one to many în care un contract are o singură casă de discuri iar casele de discuri pot să aibă mai multe contracte.

Transformarea în FN3 va arăta astfel:

Tabela CONTRACT

ID_CONTRACT	ID_MUZICIAN	NUME_CASA_DISCURI	LOCATIE_CASA_DISCURI	WEBSITE_CASA_DISCURI	AN_INFIINTARE_CASA_DISCURI	DATA_SEMNARE_CONTRACT	DATA_EXPIRARE_CONTRACT
1	17	"Reprise Records"	-	warnerrecords.com	1960	2019-01-01	2021-01-01
2	1	"Eyeball Records"	New York City	https://www.eyeballrecords.org/	1995	2001-09-12	2003-09-12
3	1	"Reprise Records"	-	warnerrecords.com	1960	2009-10-31	2011-10-31



Tabela CONTRACT

ID_CONTRACT	ID_MUZICIAN	ID_CASA_DISCURI	DATA_SEMNARE	DATA_EXPIRARE
1	17	1	2019-01-01	2021-01-01
2	1	2	2001-09-12	2003-09-12
3	1	1	2009-10-31	2011-10-31

Tabela CASA\_DISCURI

ID_CASA_DISCURI	NUME	LOCATIE	WEBSITE	AN_INFIINTARE
1	"Reprise Records"	-	warnerrecords.com	1960
2	"Eyeball Records"	New York City	https://www.eyeballrecords.org/	1995

---

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

---

➤ Cod SQL

```
CREATE SEQUENCE CASA_DISCURI_SEQ START WITH 1;
CREATE SEQUENCE CONTRACT_SEQ START WITH 1;
CREATE SEQUENCE MUZICIAN_SEQ START WITH 1;
CREATE SEQUENCE TRUPA_SEQ START WITH 1;
CREATE SEQUENCE INSTRUMENT_SEQ START WITH 1;
CREATE SEQUENCE CANTEC_SEQ START WITH 1;
CREATE SEQUENCE ALBUM_SEQ START WITH 1;
CREATE SEQUENCE ANGAJAT_SEQ START WITH 1;
CREATE SEQUENCE MANAGER_TRUPA_SEQ START WITH 1;
CREATE SEQUENCE ADMINISTRATOR_CONTRACTE_SEQ START WITH 1;
CREATE SEQUENCE CERCETATOR_MARKETING_SEQ START WITH 1;
CREATE SEQUENCE INGINER_AUDIO_SEQ START WITH 1;
```

➤ Screen Prints

```
CREATE SEQUENCE CASA_DISCURI_SEQ START WITH 1;
CREATE SEQUENCE CONTRACT_SEQ START WITH 1;
CREATE SEQUENCE MUZICIAN_SEQ START WITH 1;
CREATE SEQUENCE TRUPA_SEQ START WITH 1;
CREATE SEQUENCE INSTRUMENT_SEQ START WITH 1;
CREATE SEQUENCE CANTEC_SEQ START WITH 1;
CREATE SEQUENCE ALBUM_SEQ START WITH 1;
CREATE SEQUENCE ANGAJAT_SEQ START WITH 1;
CREATE SEQUENCE MANAGER_TRUPA_SEQ START WITH 1;
CREATE SEQUENCE ADMINISTRATOR_CONTRACTE_SEQ START WITH 1;
CREATE SEQUENCE CERCETATOR_MARKETING_SEQ START WITH 1;
CREATE SEQUENCE INGINER_AUDIO_SEQ START WITH 1;
```

Serba Ioana-Raluca  
Grupa 141

Sequence CASA\_DISCURI\_SEQ created.

Sequence CONTRACT\_SEQ created.

Sequence MUZICIAN\_SEQ created.

Sequence TRUPA\_SEQ created.

Sequence INSTRUMENT\_SEQ created.

Sequence CANTEC\_SEQ created.

Sequence ALBUM\_SEQ created.

Sequence ANGAJAT\_SEQ created.

Sequence MANAGER\_TRUPA\_SEQ created.

Sequence ADMINISTRATOR\_CONTRACTE\_SEQ created.

Sequence CERCETATOR\_MARKETING\_SEQ created.

Sequence INGINER\_AUDIO\_SEQ created.

---

## 11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea

---

(minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative;  
maxim 30 de înregistrări în fiecare tabel)

### 1. Tabela CASA\_DISCURI

➤ Cod SQL

```
--11
--casa de discuri
CREATE TABLE CASA_DISCURI (
    ID_CASA_DISCURI INT DEFAULT CASA_DISCURI_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(40) NOT NULL UNIQUE,
    ADRESA VARCHAR2(255),
    WEBSITE VARCHAR2(255),
    AN_INFIINTARE NUMBER,
    CONSTRAINT VERIFICARE_AN CHECK (AN_INFIINTARE >= 1889 AND
    AN_INFIINTARE <= 2024)
);

INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)
VALUES ('Reprise Records', 'warnerrecords.com', 1960);

INSERT INTO CASA_DISCURI (NUME, ADRESA, WEBSITE, AN_INFIINTARE)
VALUES ('Eyeball Records', 'New York City',
'https://www.eyeballrecords.org/ ', 1995);

INSERT INTO CASA_DISCURI (NUME, ADRESA, WEBSITE, AN_INFIINTARE)
VALUES ('UNDF', 'Melbourne, Victoria', 'https://unfdcentral.com',
2011);

INSERT INTO CASA_DISCURI (NUME, ADRESA, WEBSITE, AN_INFIINTARE)
VALUES ('GROBB', 'Vancouver, Canada',
'http://www.friendsoftheblues.org/', 2024);

INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)
VALUES ('Mercury Records', 'https://www.mercuryrecords.com/', 1945);
```

➤ Screen Prints

```
--11
--casa de discuri
CREATE TABLE CASA_DISCURI(
    ID_CASA_DISCURI INT DEFAULT CASA_DISCURI_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(40) NOT NULL UNIQUE,
    ADRESA VARCHAR2(255),
    WEBSITE VARCHAR2(255),
    AN_INFIINTARE NUMBER,
    CONSTRAINT VERIFICARE_AN CHECK (AN_INFIINTARE >= 1889 AND AN_INFIINTARE <= 2024)
);

INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)
VALUES ('Reprise Records', 'warnerrecords.com', 1960);

INSERT INTO CASA_DISCURI (NUME, ADRESA, WEBSITE, AN_INFIINTARE)
VALUES ('Eyeball Records', 'New York City', 'https://www.eyeballrecords.org/', 1995);

INSERT INTO CASA_DISCURI (NUME, ADRESA, WEBSITE, AN_INFIINTARE)
VALUES ('UNDF', 'Melbourne, Victoria', 'https://unfdcentral.com', 2011);

INSERT INTO CASA_DISCURI (NUME, ADRESA, WEBSITE, AN_INFIINTARE)
VALUES ('GROBB', 'Vancouver, Canada', 'http://www.friendsoftheblues.org/', 2024);

INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)
VALUES ('Mercury Records', 'https://www.mercuryrecords.com/', 1945);

SELECT *
FROM CASA_DISCURI;
```

Worksheet    Query Builder

```
SELECT * FROM CASA_DISCURI;
```

Query Result x

SQL | All Rows Fetched: 5 in 0.112 seconds

ID_CASA_DISCURI	NUME	ADRESA	WEBSITE	AN_INFIINTARE
1	1 Reprise Records (null)		warnerrecords.com	1960
2	2 Eyeball Records	New York City	https://www.eyeballrecords.org/	1995
3	3 UNDF	Melbourne, Victoria	https://unfdcentral.com	2011
4	4 GROBB	Vancouver, Canada	http://www.friendsoftheblues.org/	2024
5	5 Mercury Records (null)		https://www.mercuryrecords.com/	1945

## 2. Tabela MUZICIAN

### ➤ Cod SQL

```
--muzician
CREATE TABLE MUZICIAN
(
    ID_MUZICIAN INT DEFAULT MUZICIAN_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(40) NOT NULL,
    ZI_NASTERE DATE,
    GEN VARCHAR2(20)
);

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Gerard Way', DATE '1977-04-09', 'x');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Mikey Way', DATE '1980-09-10' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Ray Toro', DATE '1977-07-15' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Frank Iero', DATE '1981-10-31', 'm');

INSERT INTO MUZICIAN (NUME, GEN)
VALUES ('Evan Nestor', 'm');

INSERT INTO MUZICIAN (NUME, GEN)
VALUES ('Kayleigh Goldsworthy', 'f');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Doctor Sung', DATE '1998-05-26' , 'alien');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Commander Meouch', DATE '1996-04-14' , 'lion');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Lord Phobos', DATE '1993-11-11' , 'cyclops');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Haave Hogan', DATE '1996-01-23' , 'robot');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Tom Cardy', DATE '1997-06-12' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Freddy Mercury', DATE '1946-09-05' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Brian May', DATE '1947-07-19' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('John Deacon', DATE '1951-09-19' , 'm');
```

```
INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Roger Taylor', DATE '1949-07-26', 'm');
```

➤ Screen Prints

```
--muzician
CREATE TABLE MUZICIAN
(
    ID_MUZICIAN INT DEFAULT MUZICIAN_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(40) NOT NULL,
    ZI_NASTERE DATE,
    GEN VARCHAR2(20)
);

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Gerard Way', DATE '1977-04-09', 'x');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Mikey Way', DATE '1980-09-10' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Ray Toro', DATE '1977-07-15' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Frank Iero', DATE '1981-10-31', 'm');

INSERT INTO MUZICIAN (NUME, GEN)
VALUES ('Evan Nestor', 'm');

INSERT INTO MUZICIAN (NUME, GEN)
VALUES ('Kayleigh Goldsworthy', 'f');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Doctor Sung', DATE '1998-05-26' , 'alien');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Commander Meouch', DATE '1996-04-14' , 'lion');
```

```
INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Lord Phobos', DATE '1993-11-11' , 'cyclops');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Haave Hogan', DATE '1996-01-23' , 'robot');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Tom Cardy', DATE '1997-06-12' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Freddy Mercury', DATE '1946-09-05' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Brian May', DATE '1947-07-19' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('John Deacon', DATE '1951-09-19' , 'm');

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Roger Taylor', DATE '1949-07-26' , 'm');
```

```
les
SELECT *
FROM MUZICIAN;
```

Query Result x

All Rows Fetched: 15 in 0.134 seconds

	ID_MUZICIAN	NUME	ZI_NASTERE	GEN
1	1	Gerard Way	09-APR-77	x
2	2	Mikey Way	10-SEP-80	m
3	3	Ray Toro	15-JUL-77	m
4	4	Frank Iero	31-OCT-81	m
5	5	Evan Nestor	(null)	m
6	6	Kayleigh Goldsworthy	(null)	f
7	7	Doctor Sung	26-MAY-98	alien
8	8	Commander Meouch	14-APR-96	lion
9	9	Lord Phobos	11-NOV-93	cyclops
10	10	Haave Hogan	23-JAN-96	robot
11	11	Tom Cardy	12-JUN-97	m
12	12	Freddy Mercury	05-SEP-46	m
13	13	Brian May	19-JUL-47	m
14	14	John Deacon	19-SEP-51	m
15	15	Roger Taylor	26-JUL-49	m

### 3. Tabela CONTRACT

#### ➤ Cod SQL

```
--contract
CREATE TABLE CONTRACT(
    ID_CONTRACT INT DEFAULT CONTRACT_SEQ.nextval PRIMARY KEY,
    ID_MUZICIAN INT,
    ID_CASA_DISCURI INT,
    DATA_SEMNARE DATE NOT NULL,
    DATA_EXPIRARE DATE NOT NULL,
    FOREIGN KEY (ID_CASA_DISCURI) REFERENCES CASA_DISCURI(ID_CASA_DISCURI),
    FOREIGN KEY (ID_MUZICIAN) REFERENCES MUZICIAN(ID_MUZICIAN)
);

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (1,1,DATE '2003-08-31',DATE '2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (2,1,DATE '2003-08-31',DATE '2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (3,1,DATE '2003-08-31',DATE '2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (4,1,DATE '2003-08-31',DATE '2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (1,2,DATE '2013-11-24',DATE '2015-11-24');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (5,3,DATE '2019-03-21',DATE '2021-03-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (6,3,DATE '2019-03-21',DATE '2021-03-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (7,4,DATE '2024-02-22',DATE '2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (8,4,DATE '2024-02-22',DATE '2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
```

```
VALUES (9, 4, DATE '2024-02-22', DATE '2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (10, 4, DATE '2024-02-22', DATE '2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (12, 5, DATE '1975-10-21', DATE '1975-10-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (13, 5, DATE '1975-10-21', DATE '1975-10-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (14, 5, DATE '1975-10-21', DATE '1975-10-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (15, 5, DATE '1975-10-21', DATE '1975-10-21');
```

## ➤ Screen Prints

The screenshot shows a SQL worksheet interface with two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the following SQL code:

```
--contract
CREATE TABLE CONTRACT(
    ID_CONTRACT INT DEFAULT CONTRACT_SEQ.nextval PRIMARY KEY,
    ID_MUZICIAN INT,
    ID_CASA_DISCURI INT,
    DATA_SEMNARE DATE NOT NULL,
    DATA_EXPIRARE DATE NOT NULL,
    FOREIGN KEY (ID_CASA_DISCURI) REFERENCES CASA_DISCURI(ID_CASA_DISCURI),
    FOREIGN KEY (ID_MUZICIAN) REFERENCES MUZICIAN(ID_MUZICIAN)
);

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (1,1,DATE '2003-08-31',DATE '2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (2,1,DATE'2003-08-31',DATE'2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (3,1,DATE'2003-08-31',DATE'2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (4,1,DATE'2003-08-31',DATE'2006-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (1,2,DATE'2013-11-24',DATE'2015-11-24');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (5,3,DATE'2019-03-21',DATE'2021-03-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (6,3,DATE'2019-03-21',DATE'2021-03-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (7,4,DATE'2024-02-22',DATE'2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (8,4,DATE'2024-02-22',DATE'2027-02-22');
```

```
INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (9,4,DATE'2024-02-22',DATE'2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (10,4,DATE'2024-02-22',DATE'2027-02-22');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (12,5,DATE'1975-10-21',DATE'1975-10-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (13,5,DATE'1975-10-21',DATE'1975-10-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (14,5,DATE'1975-10-21',DATE'1975-10-21');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (15,5,DATE'1975-10-21',DATE'1975-10-21');
```

SELECT \* FROM CONTRACT;

Query Result | All Rows Fetched: 15 in 0.057 seconds

ID_CONTRACT	ID_MUZICIAN	ID_CASA_DISCURI	DATA_SEMNARE	DATA_EXPIRARE
1	1	1	1 31-AUG-03	31-AUG-06
2	2	2	1 31-AUG-03	31-AUG-06
3	3	3	1 31-AUG-03	31-AUG-06
4	4	4	1 31-AUG-03	31-AUG-06
5	5	1	2 24-NOV-13	24-NOV-15
6	6	5	3 21-MAR-19	21-MAR-21
7	7	6	3 21-MAR-19	21-MAR-21
8	8	7	4 22-FEB-24	22-FEB-27
9	9	8	4 22-FEB-24	22-FEB-27
10	10	9	4 22-FEB-24	22-FEB-27
11	11	10	4 22-FEB-24	22-FEB-27
12	12	12	5 21-OCT-75	21-OCT-75
13	13	13	5 21-OCT-75	21-OCT-75
14	14	14	5 21-OCT-75	21-OCT-75
15	15	15	5 21-OCT-75	21-OCT-75

#### 4. Tabela TRUPA

##### ➤ Cod SQL

```
--trupa
CREATE TABLE TRUPA(
    ID_TRUPA INT DEFAULT TRUPA_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(40) NOT NULL UNIQUE,
    GEN VARCHAR2(60) NOT NULL,
    DATA_INFIINTARE DATE NOT NULL
);

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('My Chemical Romance', 'Alternative rock', DATE '2001-09-18');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Gerard Way', 'Alternative rock', DATE '2012-12-20');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Frank Iero', 'Punk rock', DATE '2014-06-09');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('The Future Violents', 'Alternative rock', DATE '2018-11-13');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('TWRP', 'Funk rock', DATE '2011-04-05');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Tom Cardy', 'Musical Comedy', DATE '2023-07-06');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Queen', 'Progressive Rock', DATE '1970-02-28');
```

➤ Screen Prints

```
--trupa
CREATE TABLE TRUPA(
    ID_TRUPA INT DEFAULT TRUPA_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(40) NOT NULL UNIQUE,
    GEN VARCHAR2(60) NOT NULL,
    DATA_INFIINTARE DATE NOT NULL
);

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('My Chemical Romance','Alternative rock', DATE '2001-09-18');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Gerard Way','Alternative rock',DATE '2012-12-20');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Frank Iero','Punk rock',DATE '2014-06-09');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('The Future Violents','Alternative rock', DATE'2018-11-13');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('TWRP','Funk rock',DATE '2011-04-05');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Tom Cardy','Musical Comedy', DATE'2023-07-06');

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Queen','Progressive Rock', DATE'1970-02-28');
```

The screenshot shows a database client interface with a query editor and a results window.

Query Editor:

```
SELECT *
FROM TRUPA;
```

Results Window:

Query Result | All Rows Fetched: 7 in 0.102 seconds

	ID_TRUPA	NUME	GEN	DATA_INFIINTARE
1	1	My Chemical Romance	Alternative rock	18-SEP-01
2	2	Gerard Way	Alternative rock	20-DEC-12
3	3	Frank Iero	Punk rock	09-JUN-14
4	4	The Future Violents	Alternative rock	13-NOV-18
5	5	TWRP	Funk rock	05-APR-11
6	6	Tom Cardy	Musical Comedy	06-JUL-23
7	7	Queen	Progressive Rock	28-FEB-70

## 5. Tabela MEMBRU\_TRUPA

### ➤ Cod SQL

```
--membru trupa
CREATE TABLE MEMBRU_TRUPA(
    ID_MUZICIAN INT,
    ID_TRUPA INT,
    POZITIE VARCHAR2(40) NOT NULL,
    CONSTRAINT ID_MEMBRU_TRUPA PRIMARY KEY (ID_MUZICIAN, ID_TRUPA),
    FOREIGN KEY (ID_MUZICIAN) REFERENCES MUZICIAN(ID_MUZICIAN),
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    CONSTRAINT POZITIE_CHECK CHECK(UPPER(POZITIE) IN
        ('VOCALIST','CHITARIST','BASIST','CLAVIATURA','TOBOSAR', 'PIANIST'))
);

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (1,1,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (2,1,'Basist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (3,1,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (4,1,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (1,2,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (4,3,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (5,4,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (6,4,'Claviatura');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (7,5,'Claviatura');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (8,5,'Basist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (9,5,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (10,5,'Tobosar');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
```

```
VALUES (11, 6, 'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (12, 7, 'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (13, 7, 'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (14, 7, 'Basist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (15, 7, 'Tobosar');
```

➤ Screen Prints

```
--membru trupa
CREATE TABLE MEMBRU_TRUPA(
    ID_MUZICIAN INT,
    ID_TRUPA INT,
    POZITIE VARCHAR2(40) NOT NULL,
    CONSTRAINT ID_MEMBRU_TRUPA PRIMARY KEY (ID_MUZICIAN, ID_TRUPA),
    FOREIGN KEY (ID_MUZICIAN) REFERENCES MUZICIAN(ID_MUZICIAN),
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    CONSTRAINT POZITIE_CHECK CHECK(UPPER(POZITIE) IN
        ('VOCALIST','CHITARIST','BASIST','CLAVIATURA','TOBOSAR', 'PIANIST'))
);

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (1,1,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (2,1,'Basist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (3,1,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (4,1,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (1,2,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (4,3,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (5,4,'Chitarist');
```

```
INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (6,4,'Claviatura');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (7,5,'Claviatura');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (8,5,'Basist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (9,5,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (10,5,'Tobosar');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (11,6,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (12,7,'Vocalist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (13,7,'Chitarist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (14,7,'Basist');

INSERT INTO MEMBRU_TRUPA(ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (15,7,'Tobosar');
```

SELECT \* FROM MEMBRU\_TRUPA;

Query Result | All Rows Fetched: 17 in 0.177 seconds

ID_MUZICIAN	ID_TRUPA	POZITIE
1	1	1 Vocalist
2	2	1 Basist
3	3	1 Chitarist
4	4	1 Chitarist
5	1	2 Vocalist
6	4	3 Vocalist
7	5	4 Chitarist
8	6	4 Claviatura
9	7	5 Claviatura
10	8	5 Basist
11	9	5 Chitarist
12	10	5 Tobosar
13	11	6 Vocalist
14	12	7 Vocalist
15	13	7 Chitarist
16	14	7 Basist
17	15	7 Tobosar

## 6. Tabela INSTRUMENT

### ➤ Cod SQL

```
--instrument
CREATE TABLE INSTRUMENT (
    ID_INSTRUMENT INT DEFAULT INSTRUMENT_SEQ.nextval PRIMARY KEY,
    ID_TRUPA INT,
    NUME VARCHAR2(40) NOT NULL,
    UTILIZATOR VARCHAR2(40),
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    CONSTRAINT UTILIZATOR_CHECK CHECK (UPPER(UTILIZATOR)
    IN ('VOCALIST', 'CHITARIST', 'BASIST', 'CLAVIATURA', 'TOBOSAR', 'PIANIST'))
);

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (1, 'Gibson Les Paul Standard', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (1, 'Limited Edition Mikey Way Jazz Bass', 'Basist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (1, 'Epiphone Les Paul Custom PANSY', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (3, 'Joyriding Epiphone Wilshire Phantomatic', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (3, 'Danelectro DC-3', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (4, 'Fender Stratocaster', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (4, 'Yamaha DX100', 'Claviatura');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5, 'Roland AX-Synth', 'Claviatura');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5, 'Music Man Sterling Ray 34', 'Basist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5, 'Suhr Classic S', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5, 'Ludwig Black Oyster Pearl Kit', 'Tobosar');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7, 'Yamaha baby grand piano', 'Pianist');
```

```
INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Brian May Red Special', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Fender Precision Bass', 'Basist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Ludwig Classic in Silver Sparkle', 'Tobosar');
```

➤ Screen Prints

The screenshot shows a database worksheet interface with two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the following SQL code:

```
--instrument
CREATE TABLE INSTRUMENT (
    ID_INSTRUMENT INT DEFAULT INSTRUMENT_SEQ.nextval PRIMARY KEY,
    ID_TRUPA INT,
    NUME VARCHAR2(40) NOT NULL,
    UTILIZATOR VARCHAR2(40),
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    CONSTRAINT UTILIZATOR_CHECK CHECK (UPPER(UTILIZATOR)
    IN ('VOCALIST', 'CHITARIST', 'BASIST','CLAVIATURA','TOBOSAR','PIANIST'))
);

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (1,'Gibson Les Paul Standard','Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (1,'Limited Edition Mikey Way Jazz Bass', 'Basist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (1,'Epiphone Les Paul Custom PANSY', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (3,'Joyriding Epiphone Wilshire Phantomatic','Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (3,'Danelectro DC-3', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (4,'Fender Stratocaster', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (4,'Yamaha DX100', 'Claviatura');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5,'Roland AX-Synth', 'Claviatura');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5,'Music Man Sterling Ray 34', 'Basist');
```

```
INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5,'Suhr Classic S', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (5,'Ludwig Black Oyster Pearl Kit', 'Tobosar');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Yamaha baby grand piano', 'Pianist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Brian May Red Special', 'Chitarist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Fender Precision Bass', 'Basist');

INSERT INTO INSTRUMENT (ID_TRUPA, NUME, UTILIZATOR)
VALUES (7,'Ludwig Classic in Silver Sparkle', 'Tobosar');
```

```
SELECT *
FROM INSTRUMENT;
```

Query Result | All Rows Fetched: 15 in 0.037 seconds

The screenshot shows a MySQL Workbench interface with a query editor and a results viewer. The query editor contains the SQL code: 'SELECT \* FROM INSTRUMENT;'. The results viewer, titled 'Query Result', displays 15 rows of data from the INSTRUMENT table. The columns are labeled: ID\_INSTRUMENT, ID\_TRUPA, NUME, and UTILIZATOR. The data includes various musical instruments and their users.

ID_INSTRUMENT	ID_TRUPA	NUME	UTILIZATOR
1	1	1 Gibson Les Paul Standard	Chitarist
2	2	1 Limited Edition Mikey Way Jazz Bass	Basist
3	3	1 Epiphone Les Paul Custom PANSY	Chitarist
4	4	3 Joyriding Epiphone Wilshire Phantomatic	Chitarist
5	5	3 Danelectro DC-3	Chitarist
6	6	4 Fender Stratocaster	Chitarist
7	7	4 Yamaha DX100	Claviatura
8	8	5 Roland AX-Synth	Claviatura
9	9	5 Music Man Sterling Ray 34	Basist
10	10	5 Suhr Classic S	Chitarist
11	11	5 Ludwig Black Oyster Pearl Kit	Tobosar
12	12	7 Yamaha baby grand piano	Pianist
13	13	7 Brian May Red Special	Chitarist
14	14	7 Fender Precision Bass	Basist
15	15	7 Ludwig Classic in Silver Sparkle	Tobosar

## 7. Tabela CANTEC

### ➤ Cod SQL

```
--canteccanteccCREATE TABLE CANTEC(  
    ID_CANTEC INT DEFAULT CANTEC_SEQ.nextval PRIMARY KEY,  
    NUME VARCHAR2(255) NOT NULL,  
    LUNGIME NUMBER(4,2) NOT NULL,  
    CONSTRAINT LUNGIME_CHECK CHECK ( LUNGIME > 0 )  
);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Helena',3.24);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('I'm Not Okay(I Promise)',3.06);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('The Ghost Of You', 3.14);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Interlude',0.57);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Thank You for the Venom',3.41);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('To the End',3.01);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('I Never Told You What I Do for a Living',3.52);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('No Shows',4.12);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Violence',3.53);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Sewerwolf',3.50);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Record Ender',6.37);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Digital Nightmare',2.14);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Content 4 U',4.29);  
  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('HTMLOVE',4.48);
```

Şerba Ioana-Raluca  
Grupa 141

```
INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Online',4.14);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Dance Floor Cold War',3.56);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Firewall',4.00);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('A Human''s Touch',4.51);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('We Will Rock You',2.02);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('We Are The Champions',2.59);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Spread Your Wings',4.34);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Killer Queen',3.00);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Bohemian Rhapsody',5.54);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('I Want To Break Free',4.18);
```

➤ Screen Prints

```
--cantec
CREATE TABLE CANTEC(
    ID_CANTEC INT DEFAULT CANTEC_SEQ.nextval PRIMARY KEY,
    NUME VARCHAR2(255) NOT NULL,
    LUNGIME NUMBER(4,2) NOT NULL,
    CONSTRAINT LUNGIME_CHECK CHECK ( LUNGIME > 0 )
);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Helena',3.24);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('I''m Not Okay(I Promise)',3.06);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('The Ghost Of You', 3.14);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Interlude',0.57);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Thank You for the Venom',3.41);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('To the End',3.01);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('I Never Told You What I Do for a Living',3.52);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('No Shows',4.12);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Violence',3.53);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Sewerwolf',3.50);
```

Worksheet      Query Builder

```
INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Record Ender',6.37);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Digital Nightmare',2.14);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Content 4 U',4.29);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('HTMLOVE',4.48);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Online',4.14);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Dance Floor Cold War',3.56);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Firewall',4.00);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('A Human''s Touch',4.51);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('We Will Rock You',2.02);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('We Are The Champions',2.59);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Spread Your Wings',4.34);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Killer Queen',3.00);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Bohemian Rhapsody',5.54);

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('I Want To Break Free',4.18);
```

Worksheet    Query Builder

```
SELECT * FROM CANTEC;
```

Query Result x

SQL | All Rows Fetched: 24 in 0.122 seconds

	ID_CANTEC	NUME	LUNGIME
1	1	Helena	3.24
2	2	I'm Not Okay(I Promise)	3.06
3	3	The Ghost Of You	3.14
4	4	Interlude	0.57
5	5	Thank You for the Venom	3.41
6	6	To the End	3.01
7	7	I Never Told You What I Do for a Living	3.52
8	8	No Shows	4.12
9	9	Violence	3.53
10	10	Sewerwolf	3.5
11	11	Record Ender	6.37
12	12	Digital Nightmare	2.14
13	13	Content 4 U	4.29
14	14	HTMLOVE	4.48
15	15	Online	4.14
16	16	Dance Floor Cold War	3.56
17	17	Firewall	4
18	18	A Human's Touch	4.51
19	19	We Will Rock You	2.02
20	20	We Are The Champions	2.59
21	21	Spread Your Wings	4.34
22	22	Killer Queen	3
23	23	Bohemian Rhapsody	5.54
24	24	I Want To Break Free	4.18

## 8. Tabela TRUPA\_CANTEC

### ➤ Cod SQL

```
--trupa cantece
CREATE TABLE TRUPA_CANTEC (
    ID_TRUPA INT,
    ID_CANTEC INT,
    CONSTRAINT ID_TRUPA_CANTEC PRIMARY KEY (ID_TRUPA, ID_CANTEC),
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    FOREIGN KEY (ID_CANTEC) REFERENCES CANTEC(ID_CANTEC)
);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,1);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,2);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,3);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,4);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,5);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,6);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,7);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (2,8);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (3,9);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (4,9);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (3,10);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (4,10);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (4,11);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,12);
```

```
INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,13);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,14);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,15);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (6,15);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,16);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,17);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,18);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,19);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,20);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,21);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,22);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,23);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,24);
```

➤ Screen Prints

```
--trupa cantec
CREATE TABLE TRUPA_CANTEC(
    ID_TRUPA INT,
    ID_CANTEC INT,
    CONSTRAINT ID_TRUPA_CANTEC PRIMARY KEY (ID_TRUPA, ID_CANTEC),
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    FOREIGN KEY (ID_CANTEC) REFERENCES CANTEC(ID_CANTEC)
);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,1);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,2);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,3);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,4);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,5);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,6);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (1,7);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (2,8);
```

```
INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (3,9);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (4,9);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (3,10);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (4,10);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (4,11);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,12);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,13);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,14);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,15);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (6,15);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,16);
```

```
INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,17);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (5,18);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,19);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,20);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,21);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,22);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,23);

INSERT INTO TRUPA_CANTEC(ID_TRUPA, ID_CANTEC)
VALUES (7,24);
```

Worksheet    Query Builder

```
SELECT * FROM TRUPA_CANTEC;
```

Query Result x

SQL | All Rows Fetched: 27 in 0.058 seconds

	ID_TRUPA	ID_CANTEC
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6
7	1	7
8	2	8
9	3	9
10	3	10
11	4	9
12	4	10
13	4	11
14	5	12
15	5	13
16	5	14
17	5	15
18	5	16
19	5	17
20	5	18
21	6	15
22	7	19
23	7	20
24	7	21
25	7	22
26	7	23
27	7	24

## 9. Tabela ALBUM

### ➤ Cod SQL

```
--album
CREATE TABLE ALBUM(
    ID_ALBUM INT DEFAULT ALBUM_SEQ.nextval PRIMARY KEY,
    ID_TRUPA INT,
    ID_CASA_DISCURI INT,
    NUME VARCHAR2(255) NOT NULL,
    TIP VARCHAR2(10) NOT NULL,
    DATA_LANSARE DATE,
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    FOREIGN KEY (ID_CASA_DISCURI) REFERENCES
CASA_DISCURI(ID_CASA_DISCURI),
    CONSTRAINT TIP_CHECK CHECK ( UPPER(TIP) IN ('EP', 'ALBUM', 'SINGLE') )
);

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(1,1,'Helena','Single', DATE '2004-04-16');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(1,1,'Three Cheers For Sweet Revenge','Album', DATE '2004-06-08');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(2,2,'No Shows','Single', DATE '2014-08-19');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(4,3,'Barriers','EP', DATE '2019-03-31');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(5,4,'Digital Nightmare','Album', DATE '2024-04-22');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(5,4,'Online','Single', DATE '2024-03-31');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(7,5,'News Of The World','EP', DATE '1977-10-28');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(7,5,'Best Hits','Album', DATE '2001-09-13');
```

➤ Screen Prints

```
--album
CREATE TABLE ALBUM(
    ID_ALBUM INT DEFAULT ALBUM_SEQ.nextval PRIMARY KEY,
    ID_TRUPA INT,
    ID_CASA_DISCURI INT,
    NUME VARCHAR2(255) NOT NULL,
    TIP VARCHAR2(10) NOT NULL,
    DATA_LANSARE DATE,
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    FOREIGN KEY (ID_CASA_DISCURI) REFERENCES CASA_DISCURI(ID_CASA_DISCURI),
    CONSTRAINT TIP_CHECK CHECK ( UPPER(TIP) IN ('EP','ALBUM','SINGLE') )
);

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(1,1,'Helena','Single', DATE '2004-04-16');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(1,1,'Three Cheers For Sweet Revenge','Album', DATE '2004-06-08');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(2,2,'No Shows','Single', DATE '2014-08-19');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(4,3,'Barriers','EP', DATE '2019-03-31');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(5,4,'Digital Nightmare','Album', DATE '2024-04-22');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(5,4,'Online','Single', DATE '2024-03-31');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(7,5,'News Of The World','EP', DATE '1977-10-28');

INSERT INTO ALBUM(ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES(7,5,'Best Hits','Album', DATE '2001-09-13');
```

SELECT \* FROM ALBUM;

Query Result x

All Rows Fetched: 8 in 0.01 seconds

ID_ALBUM	ID_TRUPA	ID_CASA_DISCURI	NUME	TIP	DATA_LANSARE
1	1	1	1 Helena	Single	16-APR-04
2	2	1	1 Three Cheers For Sweet Revenge	Album	08-JUN-04
3	3	2	2 No Shows	Single	19-AUG-14
4	4	4	3 Barriers	EP	31-MAR-19
5	5	5	4 Digital Nightmare	Album	22-APR-24
6	6	5	4 Online	Single	31-MAR-24
7	7	7	5 News Of The World	EP	28-OCT-77
8	8	7	5 Best Hits	Album	13-SEP-01

## 10. Tabela CANTEC\_ALBUM

### ➤ Cod SQL

```
-----  
--cantec_album  
CREATE TABLE CANTEC_ALBUM (  
    ID_ALBUM,  
    ID_CANTEC,  
    CONSTRAINT ID_CANTEC_ALBUM PRIMARY KEY (ID_ALBUM, ID_CANTEC),  
    FOREIGN KEY (ID_CANTEC) REFERENCES CANTEC(ID_CANTEC),  
    FOREIGN KEY (ID_ALBUM) REFERENCES ALBUM(ID_ALBUM)  
);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(1,1);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,1);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,2);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,3);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,4);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,5);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,6);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(2,7);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(3,8);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(4,9);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(4,10);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(4,11);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(5,12);  
  
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)  
VALUES(5,13);
```

Şerba Ioana-Raluca  
Grupa 141

```
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,14);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,15);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,16);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,17);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,18);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(6,15);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(7,19);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(7,20);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(7,21);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,19);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,20);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,22);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,23);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,24);
```

➤ Screen Prints

```
--cantec_album
CREATE TABLE CANTEC_ALBUM(
    ID_ALBUM,
    ID_CANTEC,
    CONSTRAINT ID_CANTEC_ALBUM PRIMARY KEY (ID_ALBUM, ID_CANTEC),
    FOREIGN KEY (ID_CANTEC) REFERENCES CANTEC(ID_CANTEC),
    FOREIGN KEY (ID_ALBUM) REFERENCES ALBUM(ID_ALBUM)
);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (1,1);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,1);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,2);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,3);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,4);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,5);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,6);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (2,7);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (3,8);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES (4,9);
```

```
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(4,10);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(4,11);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,12);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,13);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,14);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,15);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,16);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,17);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(5,18);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC) |
VALUES(6,15);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(7,19);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(7,20);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(7,21);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,19);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,20);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,22);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,23);

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(8,24);
```

SELECT \* FROM CANTEC\_ALBUM;

Script Output x Query Result x

SQL | All Rows Fetched: 28 in 0.007 seconds

ID_ALBUM	ID_CANTEC
1	1
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	3
10	4
11	4
12	4
13	5
14	5
15	5
16	5
17	5
18	5
19	5
20	6
21	7
22	7
23	7
24	8
25	8
26	8
27	8
28	8

## 11. Tabela ANGAJAT

### ➤ Cod SQL

```
--  
--angajat  
  
CREATE TABLE ANGAJAT(  
    ID_ANGAJAT INT DEFAULT ANGAJAT_SEQ.nextval PRIMARY KEY,  
    ID_CASA_DISCURI INT,  
    NUME VARCHAR2(40) NOT NULL,  
    SALARIU NUMBER NOT NULL,  
    DEPARTAMENT VARCHAR2(40) NOT NULL,  
    DATA_ANGAJARE DATE NOT NULL,  
    FOREIGN KEY (ID_CASA_DISCURI) REFERENCES CASA_DISCURI(ID_CASA_DISCURI),  
    CONSTRAINT SALARIU_CHECK CHECK(SALARIU > 0)  
);  
  
--band managers  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (1,'Brian Schechter',10500,'Management',DATE '2002-07-09');--1  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (2,'Lola Bandit',9444,'Management',DATE '2009-08-19');--2  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (3,'Tucker Rule',6500,'Management',DATE '2016-10-20');--4  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (4,'Dylan Dove',7900,'Management',DATE '2020-06-21');--5  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (5,'Jim Beach',35800,'Management',DATE '1989-01-13');--7  
  
--administratori contracte  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (1, 'Harley Quinn', 3200, 'Administrativ', DATE '2018-05-14');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (2, 'Geralt Rivia', 8300, 'Administrativ', DATE '2000-01-19');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (3, 'Sam Winchester', 15800, 'Administrativ', DATE '2011-05-21');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (4, 'Zane Icero', 6900, 'Administrativ', DATE '2016-02-09');
```

```
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (5, 'John Doe', 4500, 'Administrativ', DATE '2003-03-22');  
  
--cercetator marketing  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (1, 'Jane Doe', 6700, 'Marketing', DATE '2019-09-10');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (2, 'Tim Curry', 9000, 'Marketing', DATE '2020-10-14');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (3, 'Kai Firre', 7900, 'Marketing', DATE '2022-03-05');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (4, 'Miguel O''hara', 6300, 'Marketing', DATE '1996-07-17');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (5, 'Cardinal Copia', 7700, 'Marketing', DATE '1989-08-05');  
  
--inginer audio  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (4, 'Miles Morales', 6600, 'Audio', DATE '2014-06-16');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (2, 'Cass Tiel', 8400, 'Audio', DATE '2014-09-19');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (3, 'Cole Wearth', 5500, 'Audio', DATE '2002-04-15');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (5, 'Peter Parker', 7400, 'Audio', DATE '2016-11-25');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (1, 'Dean Winchester', 5300, 'Audio', DATE '2012-07-07');  
  
--others  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (5, 'Bobby Singer', 6700, 'Productie', DATE '2018-12-03');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (4, 'Nya Walter', 7200, 'Productie', DATE '2015-11-11');
```

```
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (2, 'Jay Litting', 8500, 'Vanzari', DATE '2017-01-30');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (1, 'Clark Kent', 5100, 'Financiar', DATE '2014-04-01');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,  
DATA_ANGAJARE)  
VALUES (3, 'Bruce Wayne', 9200, 'Financiar', DATE '1999-02-28');
```

➤ Screen Prints

```
--angajat  
  
CREATE TABLE ANGAJAT(  
    ID_ANGAJAT INT DEFAULT ANGAJAT_SEQ.nextval PRIMARY KEY,  
    ID_CASA_DISCURI INT,  
    NUME VARCHAR2(40) NOT NULL,  
    SALARIU NUMBER NOT NULL,  
    DEPARTAMENT VARCHAR2(40) NOT NULL,  
    DATA_ANGAJARE DATE NOT NULL,  
    FOREIGN KEY (ID_CASA_DISCURI) REFERENCES CASA_DISCURI(ID_CASA_DISCURI),  
    CONSTRAINT SALARIU_CHECK CHECK(SALARIU > 0)  
);  
  
--band managers  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (1,'Brian Schechter',10500,'Management',DATE '2002-07-09');--1  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (2,'Lola Bandit',9444,'Management',DATE '2009-08-19');--2  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (3,'Tucker Rule',6500,'Management',DATE '2016-10-20');--4  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (4,'Dylan Dove',7900,'Management',DATE '2020-06-21');--5  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (5,'Jim Beach',35800,'Management',DATE '1989-01-13');--7  
  
--administratori contracte  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (1, 'Harley Quinn', 3200, 'Administrativ', DATE '2018-05-14');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (2, 'Geralt Rivia', 8300, 'Administrativ', DATE '2000-01-19');  
  
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)  
VALUES (3, 'Sam Winchester', 15800, 'Administrativ', DATE '2011-05-21');
```

Worksheet | Query Builder

```
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (4, 'Zane Icero', 6900, 'Administrativ', DATE '2016-02-09');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (5, 'John Doe', 4500, 'Administrativ', DATE '2003-03-22');

--cercetator marketing
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (1, 'Jane Doe', 6700, 'Marketing', DATE '2019-09-10');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (2, 'Tim Curry', 9000, 'Marketing', DATE '2020-10-14');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (3, 'Kai Firre', 7900, 'Marketing', DATE '2022-03-05');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (4, 'Miguel O''hara', 6300, 'Marketing', DATE '1996-07-17');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (5, 'Cardinal Copia', 7700, 'Marketing', DATE '1989-08-05');

--inginer audio
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (4, 'Miles Morales', 6600, 'Audio', DATE '2014-06-16');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (2, 'Cass Tiel', 8400, 'Audio', DATE '2014-09-19');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (3, 'Cole Wearth', 5500, 'Audio', DATE '2002-04-15');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (5, 'Peter Parker', 7400, 'Audio', DATE '2016-11-25');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (1, 'Dean Winchester', 5300, 'Audio', DATE '2012-07-07');
```

```
--others
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (5, 'Bobby Singer', 6700, 'Productie', DATE '2018-12-03');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (4, 'Nya Walter', 7200, 'Productie', DATE '2015-11-11');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (2, 'Jay Litting', 8500, 'Vanzari', DATE '2017-01-30');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (1, 'Clark Kent', 5100, 'Financiar', DATE '2014-04-01');

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (3, 'Bruce Wayne', 9200, 'Financiar', DATE '1999-02-28');
```

Worksheet    Query Builder

```
SELECT * FROM ANGAJAT;
```

Query Result x

SQL | All Rows Fetched: 25 in 0.041 seconds

ID_ANGAJAT	ID_CASA_DISCURI	NUME	SALARIU	DEPARTAMENT	DATA_ANGAJARE
1	1	Brian Schechter	10500	Management	09-JUL-02
2	2	Lola Bandit	9444	Management	19-AUG-09
3	3	Tucker Rule	6500	Management	20-OCT-16
4	4	Dylan Dove	7900	Managment	21-JUN-20
5	5	Jim Beach	35800	Management	13-JAN-89
6	6	Harley Quinn	3200	Administrativ	14-MAY-18
7	7	Geralt Rivia	8300	Administrativ	19-JAN-00
8	8	Sam Winchester	15800	Administrativ	21-MAY-11
9	9	Zane Icero	6900	Administrativ	09-FEB-16
10	10	John Doe	4500	Administrativ	22-MAR-03
11	11	Jane Doe	6700	Marketing	10-SEP-19
12	12	Tim Curry	9000	Marketing	14-OCT-20
13	13	Kai Firre	7900	Marketing	05-MAR-22
14	14	Miguel O'hara	6300	Marketing	17-JUL-96
15	15	Cardinal Copia	7700	Marketing	05-AUG-89
16	16	Miles Morales	6600	Audio	16-JUN-14
17	17	Cass Tiel	8400	Audio	19-SEP-14
18	18	Cole Wearth	5500	Audio	15-APR-02
19	19	Peter Parker	7400	Audio	25-NOV-16
20	20	Dean Winchester	5300	Audio	07-JUL-12
21	21	Bobby Singer	6700	Productie	03-DEC-18
22	22	Nya Walter	7200	Productie	11-NOV-15
23	23	Jay Litting	8500	Vanzari	30-JAN-17
24	24	Clark Kent	5100	Financiar	01-APR-14
25	25	Bruce Wayne	9200	Financiar	28-FEB-99

## 12. Tabela MANAGER\_TRUPA

### ➤ Cod SQL

```
-----  
--manager trupa  
CREATE TABLE MANAGER_TRUPA(  
    ID_MANAGER_TRUPA INT DEFAULT MANAGER_TRUPA_SEQ.nextval PRIMARY KEY,  
    ID_TRUPA INT,  
    ID_ANGAJAT INT,  
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),  
    FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT)  
);  
  
INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)  
VALUES (1,1);  
  
INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)  
VALUES (2,2);  
  
INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)  
VALUES (4,3);  
  
INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)  
VALUES (5,4);  
  
INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)  
VALUES (7,5);
```

➤ Screen Prints

```
--manager trupa
CREATE TABLE MANAGER_TRUPA(
    ID_MANAGER_TRUPA INT DEFAULT MANAGER_TRUPA_SEQ.nextval PRIMARY KEY,
    ID_TRUPA INT,
    ID_ANGAJAT INT,
    FOREIGN KEY (ID_TRUPA) REFERENCES TRUPA(ID_TRUPA),
    FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT)
);

INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)
VALUES (1,1);

INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)
VALUES (2,2);

INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)
VALUES (4,3);

INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)
VALUES (5,4);

INSERT INTO MANAGER_TRUPA(ID_TRUPA, ID_ANGAJAT)
VALUES (7,5);
```

SELECT \* FROM MANAGER\_TRUPA;

Script Output x | Query Result x | SQL | All Rows Fetched: 5 in 0.052 seconds

ID_MANAGER_TRUPA	ID_TRUPA	ID_ANGAJAT
1	1	1
2	2	2
3	3	4
4	4	5
5	5	7

### 13. Tabela ADMINISTRATOR\_CONTRACTE

#### ➤ Cod SQL

```
-----  
--administrator contracte  
CREATE TABLE ADMINISTRATOR_CONTRACTE (  
    ID_ADMINISTRATOR_CONTRACTE INT DEFAULT  
ADMINISTRATOR_CONTRACTE_SEQ.nextval PRIMARY KEY,  
    ID_ANAJAT INT,  
    SPECIALITATE VARCHAR2(40),  
    FOREIGN KEY (ID_ANAJAT) REFERENCES ANAJAT(ID_ANAJAT)  
);  
  
INSERT INTO ADMINISTRATOR_CONTRACTE (ID_ANAJAT, SPECIALITATE)  
VALUES (6, 'Vânzări');  
  
INSERT INTO ADMINISTRATOR_CONTRACTE (ID_ANAJAT, SPECIALITATE)  
VALUES (7, 'Angajări');  
  
INSERT INTO ADMINISTRATOR_CONTRACTE (ID_ANAJAT, SPECIALITATE)  
VALUES (8, 'Parteneriate');  
  
INSERT INTO ADMINISTRATOR_CONTRACTE (ID_ANAJAT, SPECIALITATE)  
VALUES (9, 'Publicări');  
  
INSERT INTO ADMINISTRATOR_CONTRACTE (ID_ANAJAT)  
VALUES (10);
```

➤ Screen Prints

```
--administrator contracte
CREATE TABLE ADMINISTRATOR_CONTRACTE(
    ID_ADMINISTRATOR_CONTRACTE INT DEFAULT ADMINISTRATOR_CONTRACTE_SEQ.nextval PRIMARY KEY,
    ID_ANAJAT INT,
    SPECIALITATE VARCHAR2(40),
    FOREIGN KEY (ID_ANAJAT) REFERENCES ANAJAT(ID_ANAJAT)
);

INSERT INTO ADMINISTRATOR_CONTRACTE(ID_ANAJAT, SPECIALITATE)
VALUES (6,'Vânzări');

INSERT INTO ADMINISTRATOR_CONTRACTE(ID_ANAJAT, SPECIALITATE)
VALUES (7,'Angajări');

INSERT INTO ADMINISTRATOR_CONTRACTE(ID_ANAJAT, SPECIALITATE)
VALUES (8,'Parteneriate');

INSERT INTO ADMINISTRATOR_CONTRACTE(ID_ANAJAT, SPECIALITATE)
VALUES (9,'Publicări');

INSERT INTO ADMINISTRATOR_CONTRACTE(ID_ANAJAT)
VALUES (10);
```

SELECT \* FROM ADMINISTRATOR\_CONTRACTE;

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.184 seconds

ID_ADMINISTRATOR_CONTRACTE	ID_ANAJAT	SPECIALITATE
1	1	6 Vânzări
2	2	7 Angajări
3	3	8 Parteneriate
4	4	9 Publicări
5	5	10 (null)

## 14. Tabela CERCETATOR\_MARKETING

### ➤ Cod SQL

```
-----  
--cercetator marketing  
CREATE TABLE CERCETATOR_MARKETING(  
    ID_CERCETATOR_MARKETING INT DEFAULT CERCETATOR_MARKETING_SEQ.nextval  
PRIMARY KEY,  
    ID_ANGAJAT INT,  
    CATEGORIE VARCHAR2(40),  
    FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT)  
) ;  
  
INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)  
VALUES (11, 'Social Media');  
  
INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)  
VALUES (11, 'Generational');  
  
INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT)  
VALUES (11);  
  
INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)  
VALUES (11, 'Inter-Platformer');  
  
INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)  
VALUES (11, 'Surveyor');
```

➤ Screen Prints

```
--cercetator marketing
CREATE TABLE CERCETATOR_MARKETING(
    ID_CERCETATOR_MARKETING INT DEFAULT CERCETATOR_MARKETING_SEQ.nextval PRIMARY KEY,
    ID_ANGAJAT INT,
    CATEGORIE VARCHAR2(40),
    FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT)
);

INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)
VALUES (11,'Social Media');

INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)
VALUES (12,'Generational');

INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT)
VALUES (13);

INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)
VALUES (14,'Inter-Platformer');

INSERT INTO CERCETATOR_MARKETING(ID_ANGAJAT, CATEGORIE)
VALUES (15,'Surveyor');
```

```
SELECT * FROM CERCETATOR_MARKETING;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 5 in 0.066 seconds

	ID_CERCETATOR_MARKETING	ID_ANGAJAT	CATEGORIE
1		1	11 Social Media
2		2	12 Generational
3		3	13 (null)
4		4	14 Inter-Platformer
5		5	15 Surveyor

## 15. Tabela INGINER\_AUDIO

### ➤ Cod SQL

```
--inginer audio
CREATE TABLE INGINER_AUDIO(
    ID_INGINER_AUDIO INT DEFAULT INGINER_AUDIO_SEQ.nextval PRIMARY KEY,
    ID_ANGAJAT INT,
    SPECIALITATE_GEN VARCHAR2(40),
    FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJAT(ID_ANGAJAT)
);

INSERT INTO INGINER_AUDIO(ID_ANGAJAT, SPECIALITATE_GEN)
VALUES (16, 'Hip Hop');

INSERT INTO INGINER_AUDIO(ID_ANGAJAT, SPECIALITATE_GEN)
VALUES (17, 'Punk Rock');

INSERT INTO INGINER_AUDIO(ID_ANGAJAT, SPECIALITATE_GEN)
VALUES (18, 'Alternative Rock');

INSERT INTO INGINER_AUDIO(ID_ANGAJAT, SPECIALITATE_GEN)
VALUES (19, 'Funk Rock');

INSERT INTO INGINER_AUDIO(ID_ANGAJAT, SPECIALITATE_GEN)
VALUES (20, 'Progressive Rock');
```

➤ Screen prints

Worksheet    Query Builder

```
--inginer audio
CREATE TABLE INGINER_AUDIO(
    ID_INGINER_AUDIO INT DEFAULT INGINER_AUDIO_SEQ.nextval PRIMARY KEY,
    ID_ANAJAT INT,
    SPECIALITATE_GEN VARCHAR2(40),
    FOREIGN KEY (ID_ANAJAT) REFERENCES ANGAJAT(ID_ANAJAT)
);

INSERT INTO INGINER_AUDIO(ID_ANAJAT, SPECIALITATE_GEN)
VALUES (16, 'Hip Hop');

INSERT INTO INGINER_AUDIO(ID_ANAJAT, SPECIALITATE_GEN)
VALUES (17, 'Punk Rock');

INSERT INTO INGINER_AUDIO(ID_ANAJAT, SPECIALITATE_GEN)
VALUES (18, 'Alternative Rock');

INSERT INTO INGINER_AUDIO(ID_ANAJAT, SPECIALITATE_GEN)
VALUES (19, 'Funk Rock');

INSERT INTO INGINER_AUDIO(ID_ANAJAT, SPECIALITATE_GEN)
VALUES (20, 'Progressive Rock');

SELECT * FROM INGINER_AUDIO;
```

Script Output X    Query Result X

SQL | All Rows Fetched: 5 in 0.13 seconds

ID_INGINER_AUDIO	ID_ANAJAT	SPECIALITATE_GEN
1	1	16 Hip Hop
2	2	17 Punk Rock
3	3	18 Alternative Rock
4	4	19 Funk Rock
5	5	20 Progressive Rock

---

**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, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (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)

➤ Cererea 1

➤ Cerința în limbaj natural:

Afișați detalii despre albumul (numele, anul în care a apărut, durata întreaga a sa în minute și numele casei de discuri a cărei aparține) (de tip Album) cu cea mai lungă durată care are minim 6 cântece, alături de durata medie a unui album (de tip Album).

➤ Cod SQL

```
WITH ALBUM_DURATA AS --bloc de cerere WITH
  (SELECT A.ID_ALBUM, C.DURATA
   FROM ALBUM A
   JOIN (
     SELECT CA.ID_ALBUM, SUM(C.LUNGIME) AS DURATA
     FROM CANTEC C
     JOIN CANTEC_ALBUM CA ON C.ID_CANTEC = CA.ID_CANTEC
     GROUP BY CA.ID_ALBUM
     HAVING COUNT(CA.ID_ALBUM)>=6) C ON C.ID_ALBUM = A.ID_ALBUM
   -- funcții grup, filtrare la nivel de grupuri cu subcereri
   -- nesincronizate
   WHERE UPPER(A.TIP) = 'ALBUM' --functie pe sir de caractere
  )
  SELECT *
  FROM (
```

```
SELECT ALBM.NUME, ALBM.AN_LANSARE, AL.DURATA, ALBM.CASA_DISCURI,
       (SELECT ROUND(AVG(DURATA), 2)
        FROM ALBUM_DURATA) AS DURATA_MEDIE
  FROM (--subcerere nesincronizata în clauza FROM
        SELECT A.ID_ALBUM, TO_CHAR(A.DATA_LANSARE, 'YYYY') AS AN_LANSARE,
               A.NUME, D.NUME AS CASA_DISCURI
        FROM ALBUM A-- ^ functie pe data calendaristica
        JOIN CASA_DISCURI D ON A.ID_CASA_DISCURI = D.ID_CASA_DISCURI)
ALBM
JOIN ALBUM_DURATA AL ON ALBM.ID_ALBUM = AL.ID_ALBUM
ORDER BY AL.DURATA DESC
) WHERE ROWNUM = 1;
```

➤ Aceasta cerere conține:

- subcereri sincronizate în care intervin 3 tabele, în blocul de cerere WITH
- grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (în cadrul aceleiași cereri, în blocul de cerere WITH)
- se utilizează 1 bloc de cerere WITH
- subcerere nesincronizată în clauza FROM
- funcție pe date calendaristice (TO\_CHAR)
- funcție pe sir de caractere (UPPER)

➤ Screen prints

```

Worksheet      Query Builder
WITH ALBUM_DURATA AS --bloc de cerere WITH
  (SELECT A.ID_ALBUM, C.DURATA
   FROM ALBUM A
   JOIN (
     SELECT CA.ID_ALBUM, SUM(C.LUNGIME) AS DURATA
     FROM CANTEC C
     JOIN CANTEC_ALBUM CA ON C.ID_CANTEC = CA.ID_CANTEC
     GROUP BY CA.ID_ALBUM
     HAVING COUNT(CA.ID_ALBUM)>=6) C ON C.ID_ALBUM = A.ID_ALBUM
   -- functii grup, filtrare la nivel de grupuri cu subcereri nesincronizate
   WHERE UPPER(A.TIP) = 'ALBUM'--functie pe sir de caractere
  )
  SELECT *
  FROM (
    SELECT ALBM.NUME, ALBM.AN_LANSARE, AL.DURATA, ALBM.CASA_DISCURI,
           (SELECT ROUND(AVG(DURATA), 2)
            FROM ALBUM_DURATA) AS DURATA_MEDIE
      FROM (--subcerere nesincronizata in clauza FROM
            SELECT A.ID_ALBUM, TO_CHAR(A.DATA_LANSARE,'YYYY') AS AN_LANSARE, A.NUME, D.NUME AS CASA_DISCURI
            FROM ALBUM A-- ^ functie pe data calendaristica
            JOIN CASA_DISCURI D ON A.ID_CASA_DISCURI = D.ID_CASA_DISCURI) ALBM
      JOIN ALBUM_DURATA AL ON ALBM.ID_ALBUM = AL.ID_ALBUM
     ORDER BY AL.DURATA DESC
  )WHERE ROWNUM = 1;
  
```

Script Output | Query Result | All Rows Fetched: 1 in 0.167 seconds

NUME	AN_LANSARE	DURATA	CASA_DISCURI	DURATA_MEDIE
1 Digital Nightmare	2024	27.12 GROBB		23.54

➤ Cererea 2

➤ Cerință în limbaj natural:

Pentru fiecare album care are măcar 10 ani vechime afișați titlul întreg de lansare  
(format din nume + '-' + tipul albumului în felul următor: dacă e Album: Full Length Album,

dacă este EP: Extended Playlist, dacă e Single: Single Hit) alături de numele trupei și numele managerului de trupă (dacă există, sau dacă `-' nu există).

➤ Cod SQL

```
SELECT CONCAT(NUME, -- FUNCTIE PE SIR concat
              CASE WHEN UPPER(TIP)='ALBUM' -- FUNCTIE PE SIR UPPER
                    THEN ' - Full Leght Album'
                    WHEN UPPER(TIP)='EP'
                    THEN ' - Extended Playlist'
                    WHEN UPPER(TIP)='SINGLE'
                    THEN ' - Single Hit' END) AS TITLU, NUME_TRUPA,
        NUME_MANAGER
     FROM (SELECT A.NUME, A.TIP, T.NUME_TRUPA, A.DATA_LANSARE,
                  NVL(T.NUME_MANAGER, '-') AS NUME_MANAGER
            FROM ALBUM A
           JOIN (
                  SELECT T.NUME AS NUME_TRUPA, T.ID_TRUPA, M.NUME AS
                      NUME_MANAGER
                 FROM (
                        SELECT ANJ.NUME, MT.ID_TRUPA
                          FROM MANAGER_TRUPA MT
                         JOIN ANGAJAT ANJ ON MT.ID_ANGAJAT = ANJ.ID_ANGAJAT) M
                RIGHT OUTER JOIN TRUPA T ON T.ID_TRUPA = M.ID_TRUPA -- ca să
        luam si trupele fara manageri
               )
              T ON A.ID_TRUPA = T.ID_TRUPA)
    WHERE MONTHS_BETWEEN(SYSDATE, DATA_LANSARE) >= 120; -- FUNCTIE PE DATA
CALENDARISTICA
--SYSDATE, SI MONTHS BETWEEN
```

➤ Aceasta cerere conține:

- o expresie de tip CASE și o expresie de tip NVL
- 2 funcții pe date calendaristice (SYSDATE, MONTHS\_BETWEEN)
- 2 funcții pe șiruri de caractere (CONCAT, UPPER)
- subcerere nesincronizată în clauza FROM
- subcereri sincronizate în care intervin 4 tabele

➤ Screen prints

The screenshot shows an Oracle SQL developer interface. The top half displays a complex SQL query. The bottom half shows the resulting table.

```
SELECT CONCAT(UME, --FUNCTIE PE SIR concat
CASE WHEN UPPER(TIP)='ALBUM'--FUNCTIE PE SIR UPPER
THEN ' - Full Leght Album'
WHEN UPPER(TIP)='EP'
THEN ' - Extended Playlist'
WHEN UPPER(TIP)='SINGLE'
THEN ' - Single Hit' END) AS TITLU, UME_TRUPA, UME_MANAGER
FROM (SELECT A.UME, A.TIP, T.UME_TRUPA, A.DATA_LANSARE, NVL(T.UME_MANAGER,'-') AS UME_MANAGER
FROM ALBUM A
JOIN (
SELECT T.UME AS UME_TRUPA, T.ID_TRUPA, M.UME AS UME_MANAGER
FROM (
SELECT ANJ.UME, MT.ID_TRUPA
FROM MANAGER_TRUPA MT
JOIN ANGAJAT ANJ ON MT.ID_ANGAJAT = ANJ.ID_ANGAJAT) M
RIGHT OUTER JOIN TRUPA T ON T.ID_TRUPA = M.ID_TRUPA -- ca sa luam si trupele fara manageri
)
T ON A.ID_TRUPA = T.ID_TRUPA)
WHERE MONTHS_BETWEEN(SYSDATE, DATA_LANSARE) >= 120; -- FUNCTIE PE DATA CALENDARISTICA
--SYSDATE, SI MONTHS BETWEEN
```

Query Result | All Rows Fetched: 4 in 1.044 seconds

TITLU	UME_TRUPA	UME_MANAGER
1 Helena - Single Hit	My Chemical Romance	Brian Schechter
2 Three Cheers For Sweet Revenge - Full Leght Album	My Chemical Romance	Brian Schechter
3 News Of The World - Extended Playlist	Queen	Jim Beach
4 Best Hits - Full Leght Album	Queen	Jim Beach

➤ Cererea 3

➤ Cerință în limbaj natural:

Afişați numele, salariul, noul salariu, departamentul și tipul de angajat astfel încât pentru noul salariu managerii de trupă primesc spor de 50%, administratorii de contracte de 40%, cercetătorii de marketing de 30%, inginerii audio de 45% iar restul angajaților de 20%. Dacă angajații nu au un tip cunoscut se va afișa "-". Ordonați după salariul nou descrescător, respectiv după nume

➤ Cod SQL

```
SELECT NUME, SALARIU,
       SALARIU*(DECODE(TIP,'Manager Trupa', 1.5,--functia decode
                        'Administrator Contracte', 1.4,
                        'Cercetator Marketing', 1.3,
                        'Inginer Audio', 1.45, 1.2))
       AS SALARIU_SPORIT, DEPARTAMENT, TIP
  FROM (SELECT A.NUME,
               A.SALARIU,
               A.DEPARTAMENT,
               NVL(CASE --functia nvl
                     WHEN MT.ID_ANGAJAT IS NOT NULL THEN 'Manager
Trupa'
                     WHEN AC.ID_ANGAJAT IS NOT NULL THEN 'Administrator
Contracte'
                     WHEN CM.ID_ANGAJAT IS NOT NULL THEN 'Cercetator
Marketing'
                     WHEN IA.ID_ANGAJAT IS NOT NULL THEN 'Inginer
Audio'
                     END, '-') AS TIP
     FROM ANGAJAT A
       LEFT OUTER JOIN MANAGER_TRUPA MT ON A.ID_ANGAJAT =
MT.ID_ANGAJAT
       LEFT OUTER JOIN ADMINISTRATOR_CONTRACTE AC ON
A.ID_ANGAJAT = AC.ID_ANGAJAT
       LEFT OUTER JOIN CERCETATOR_MARKETING CM ON A.ID_ANGAJAT
= CM.ID_ANGAJAT
       LEFT OUTER JOIN INGINER_AUDIO IA ON A.ID_ANGAJAT =
IA.ID_ANGAJAT)
 ORDER BY SALARIU_SPORIT DESC, NUME; --ordonare
```

➤ Aceasta cerere conține:

- ordonări, NVL și DECODE în cadrul aceleasi cereri
- subcerere nesincronizată în clauza FROM
- subcereri sincronizate în care intervin 5 tabele

➤ Screen prints

```

SELECT NUME, SALARIU,
       SALARIU*(DECODE(TIP,'Manager Trupa', 1.5,--functia decode
                      'Administrator Contracte', 1.4,
                      'Cercetator Marketing', 1.3,
                      'Inginer Audio', 1.45, 1.2))
  AS SALARIU_SPORIT, DEPARTAMENT, TIP
FROM (SELECT A.NUME,
            A.SALARIU,
            A.DEPARTAMENT,
            NVL(CASE --functia nvl
                  WHEN MT.ID_ANGAJAT IS NOT NULL THEN 'Manager Trupa'
                  WHEN AC.ID_ANGAJAT IS NOT NULL THEN 'Administrator Contracte'
                  WHEN CM.ID_ANGAJAT IS NOT NULL THEN 'Cercetator Marketing'
                  WHEN IA.ID_ANGAJAT IS NOT NULL THEN 'Inginer Audio'
                  END, '-') AS TIP
      FROM ANGAJAT A
      LEFT OUTER JOIN MANAGER_TRUPA MT ON A.ID_ANGAJAT = MT.ID_ANGAJAT
      LEFT OUTER JOIN ADMINISTRATOR_CONTRACTE AC ON A.ID_ANGAJAT = AC.ID_ANGAJAT
      LEFT OUTER JOIN CERCETATOR_MARKETING CM ON A.ID_ANGAJAT = CM.ID_ANGAJAT
      LEFT OUTER JOIN INGINER_AUDIO IA ON A.ID_ANGAJAT = IA.ID_ANGAJAT)
ORDER BY SALARIU_SPORIT DESC, NUME; --ordonare

```

Query Result | SQL | All Rows Fetched: 25 in 0.054 seconds

NUME	SALARIU	SALARIU_SPORIT	DEPARTAMENT	TIP
1 Jim Beach	35800	53700	Management	Manager Trupa
2 Sam Winchester	15800	22120	Administrativ	Administrator Contracte
3 Brian Schechter	10500	15750	Management	Manager Trupa
4 Lola Bandit	9444	14166	Management	Manager Trupa
5 Cass Tiel	8400	12180	Audio	Inginer Audio
6 Dylan Dove	7900	11850	Management	Manager Trupa
7 Tim Curry	9000	11700	Marketing	Cercetator Marketing
8 Geralt Rivia	8300	11620	Administrativ	Administrator Contracte
9 Bruce Wayne	9200	11040	Financiar	-
10 Peter Parker	7400	10730	Audio	Inginer Audio
11 Kai Firre	7900	10270	Marketing	Cercetator Marketing
12 Jay Litting	8500	10200	Vanzari	-
13 Cardinal Copia	7700	10010	Marketing	Cercetator Marketing
14 Tucker Rule	6500	9750	Management	Manager Trupa
15 Zane Icero	6900	9660	Administrativ	Administrator Contracte
16 Miles Morales	6600	9570	Audio	Inginer Audio
17 Jane Doe	6700	8710	Marketing	Cercetator Marketing
18 Nya Walter	7200	8640	Productie	-
19 Miguel O'hara	6300	8190	Marketing	Cercetator Marketing
20 Bobby Singer	6700	8040	Productie	-
21 Cole Wearth	5500	7975	Audio	Inginer Audio
22 Dean Winchester	5300	7685	Audio	Inginer Audio
23 John Doe	4500	6300	Administrativ	Administrator Contracte
24 Clark Kent	5100	6120	Financiar	-
25 Harley Quinn	3200	4480	Administrativ	Administrator Contracte

➤ Cerere 4

➤ Cerință în limbaj natural:

Pentru fiecare muzician afișați numele și numărul de instrumente la care acesta are acces și pe care le poate folosi. De asemenea afișați dacă acesta mai are contracte valide.

➤ Cod SQL

```
WITH IM AS (
    SELECT M.ID_MUZICIAN, COUNT(ID_INSTRUMENT) AS NR_INSTRUMENTE_ACESS
    FROM (SELECT M.ID_MUZICIAN, MT.POZITIE, MT.ID_TRUPA
          FROM MUZICIAN M
          JOIN MEMBRU_TRUPA MT ON M.ID_MUZICIAN = MT.ID_MUZICIAN) M
    LEFT OUTER JOIN (
        SELECT I.ID_INSTRUMENT, IUTILIZATOR, T.ID_TRUPA
        FROM TRUPA T
        JOIN INSTRUMENT I ON T.ID_TRUPA = I.ID_TRUPA
    ) I ON I.ID_TRUPA = M.ID_TRUPA AND IUTILIZATOR = M.POZITIE
    GROUP BY ID_MUZICIAN
)
SELECT MI.NUME, MI.NR_INSTRUMENTE_ACESS,
    CASE WHEN C.DATA_EXPIRARE < SYSDATE THEN 'CONTRACT EXPIRAT'
         ELSE 'CONTRACT VALID' END AS STARE_CONTRACT
FROM (
    SELECT M.ID_MUZICIAN, M.NUME, IM.NR_INSTRUMENTE_ACESS
    FROM MUZICIAN M
    JOIN IM ON IM.ID_MUZICIAN=M.ID_MUZICIAN
) MI
JOIN CONTRACT C ON C.ID_MUZICIAN = MI.ID_MUZICIAN;
```

➤ Aceasta cerere conține:

- se utilizeaza 1 bloc de cerere WITH
- o funcție grup
- o expresie de tip CASE
- SYSDATE, funcție de tip dată calendaristică
- subcerere nesincronizata în clauza FROM
- subcereri sincronizate în care intervin 4 tabele

➤ Screen prints

```
WITH IM AS (
    SELECT M.ID_MUZICIAN, COUNT(ID_INSTRUMENT) AS NR_INSTRUMENTE_ACESS
    FROM(SELECT M.ID_MUZICIAN, MT.POZITIE, MT.ID_TRUPA
        FROM MUZICIAN M
        JOIN MEMBRU_TRUPA MT ON M.ID_MUZICIAN = MT.ID_MUZICIAN) M
    LEFT OUTER JOIN (
        SELECT I.ID_INSTRUMENT, IUTILIZATOR, T.ID_TRUPA
        FROM TRUPA T
        JOIN INSTRUMENT I ON T.ID_TRUPA = I.ID_TRUPA
    ) I ON I.ID_TRUPA = M.ID_TRUPA AND IUTILIZATOR = M.POZITIE
    GROUP BY ID_MUZICIAN
)
SELECT MI.NUME, MI.NR_INSTRUMENTE_ACESS,
    CASE WHEN C.DATA_EXPIRARE < SYSDATE THEN 'CONTRACT EXPIRAT'
        ELSE 'CONTRACT VALID' END AS STARE_CONTRACT
FROM (
    SELECT M.ID_MUZICIAN, M.NUME, IM.NR_INSTRUMENTE_ACESS
    FROM MUZICIAN M
    JOIN IM ON IM.ID_MUZICIAN=M.ID_MUZICIAN
    ) MI
JOIN CONTRACT C ON C.ID_MUZICIAN = MI.ID_MUZICIAN;
```

Query Result

SQL | All Rows Fetched: 15 in 0.134 seconds

NUME	NR_INSTRUMENTE_ACESS	STARE_CONTRACT
1 Gerard Way		0 CONTRACT EXPIRAT
2 Mikey Way		1 CONTRACT EXPIRAT
3 Ray Toro		2 CONTRACT EXPIRAT
4 Frank Iero		2 CONTRACT EXPIRAT
5 Gerard Way		0 CONTRACT EXPIRAT
6 Evan Nestor		1 CONTRACT EXPIRAT
7 Kayleigh Goldsworthy		1 CONTRACT EXPIRAT
8 Doctor Sung		1 CONTRACT VALID
9 Commander Meouch		1 CONTRACT VALID
10 Lord Phobos		1 CONTRACT VALID
11 Haave Hogan		1 CONTRACT VALID
12 Freddy Mercury		0 CONTRACT EXPIRAT
13 Brian May		1 CONTRACT EXPIRAT
14 John Deacon		1 CONTRACT EXPIRAT
15 Roger Taylor		1 CONTRACT EXPIRAT

➤ Cererea 5

➤ Cerință în limbaj natural:

Pentru fiecare muzician afișați numărul de minute total al cântecelor la care acesta a contribuit și ordonați descrescător după acest număr

➤ Cod SQL

```
SELECT MTC.NUME, SUM(C.LUNGIME) AS MINUTE_TOTAL
FROM (
    SELECT MT2.ID_MUZICIAN, MT2.NUME, TC.ID_CANTEC
    FROM (
        SELECT M.ID_MUZICIAN, M.NUME, MTT.ID_TRUPA
        FROM MUZICIAN M
        JOIN (
            SELECT MT.ID_TRUPA, MT.ID_MUZICIAN
            FROM MEMBRU_TRUPA MT
            JOIN TRUPA T ON MT.ID_TRUPA = T.ID_TRUPA
        ) MTT ON MTT.ID_MUZICIAN = M.ID_MUZICIAN) MT2
        JOIN TRUPA_CANTEC TC ON TC.ID_TRUPA = MT2.ID_TRUPA) MTC
    JOIN CANTEC C ON MTC.ID_CANTEC = C.ID_CANTEC
    GROUP BY MTC.ID_MUZICIAN, MTC.NUME
    ORDER BY MINUTE_TOTAL DESC
```

➤ Aceasta cerere conține:

- subcereri nesincronizate în clauza FROM
- grupare
- ordonare
- intervin 4 tabele

➤ Screen prints

```
SELECT MTC.NUME, SUM(C.LUNGIME) AS MINUTE_TOTAL
FROM(
    SELECT MT2.ID_MUZICIAN, MT2.NUME, TC.ID_CANTEC
    FROM(
        SELECT M.ID_MUZICIAN, M.NUME, MTT.ID_TRUPA
        FROM MUZICIAN M
        JOIN (
            SELECT MT.ID_TRUPA, MT.ID_MUZICIAN
            FROM MEMBRU_TRUPA MT
            JOIN TRUPA T ON MT.ID_TRUPA = T.ID_TRUPA
            )MTT ON MTT.ID_MUZICIAN = M.ID_MUZICIAN) MT2
        JOIN TRUPA_CANTEC TC ON TC.ID_TRUPA = MT2.ID_TRUPA) MTC
    JOIN CANTEC C ON MTC.ID_CANTEC = C.ID_CANTEC
    GROUP BY MTC.ID_MUZICIAN, MTC.NUME
    ORDER BY MINUTE_TOTAL DESC;
```

Query Result

SQL | All Rows Fetched: 15 in 0.093 seconds

NUME	MINUTE_TOTAL
1 Haave Hogan	27.12
2 Lord Phobos	27.12
3 Commander Meouch	27.12
4 Doctor Sung	27.12
5 Frank Iero	26.98
6 Gerard Way	24.07
7 John Deacon	21.67
8 Roger Taylor	21.67
9 Brian May	21.67
10 Freddy Mercury	21.67
11 Mikey Way	19.95
12 Ray Toro	19.95
13 Evan Nestor	13.4
14 Kayleigh Goldsworthy	13.4
15 Tom Cardy	4.14

---

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

---

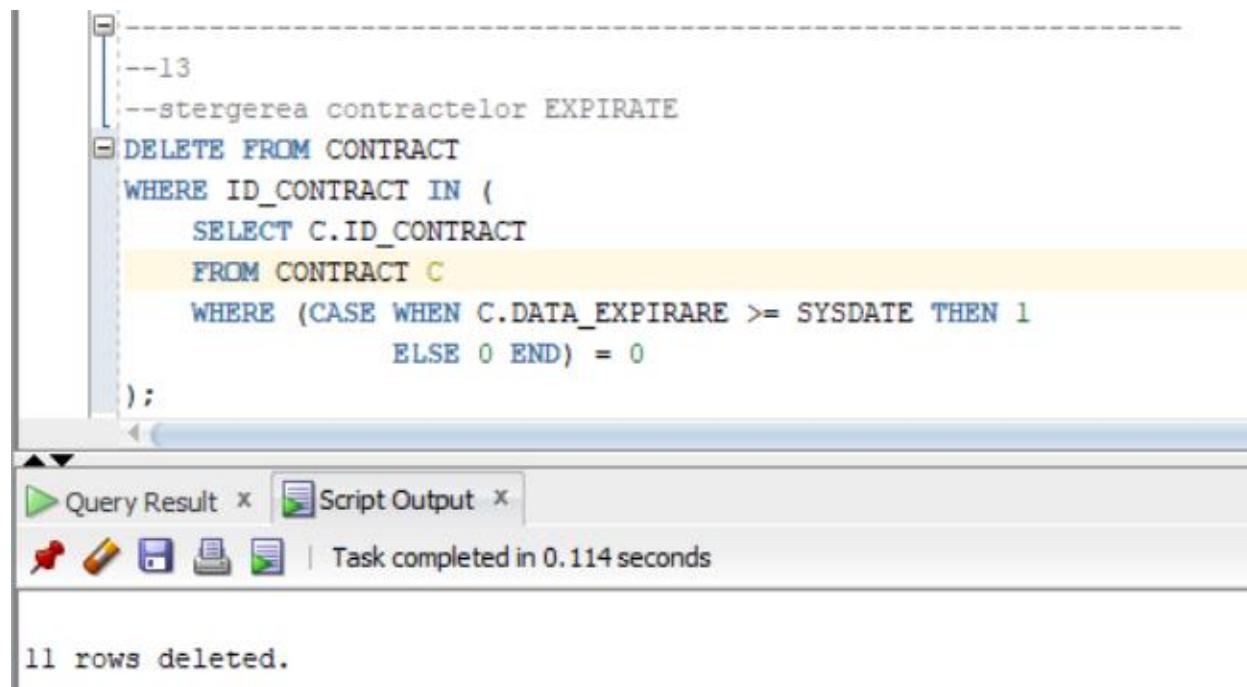
➤ Operația 1

Ștergerea contractelor expirate

➤ Cod SQL

```
DELETE FROM CONTRACT
WHERE ID_CONTRACT IN (
    SELECT C.ID_CONTRACT
    FROM CONTRACT C
    WHERE (CASE WHEN C.DATA_EXPIRARE >= SYSDATE THEN 1
               ELSE 0 END) = 0
);
```

➤ Screen Prints



The screenshot shows the Oracle SQL Developer interface. The code editor pane contains the following SQL script:

```
--13
--stergerea contractelor EXPIRATE
DELETE FROM CONTRACT
WHERE ID_CONTRACT IN (
    SELECT C.ID_CONTRACT
    FROM CONTRACT C
    WHERE (CASE WHEN C.DATA_EXPIRARE >= SYSDATE THEN 1
               ELSE 0 END) = 0
);
```

The line starting with "FROM CONTRACT C" is highlighted with a yellow background. Below the code editor is the "Script Output" tab, which displays the message "Task completed in 0.114 seconds". At the bottom of the interface, the status bar shows "11 rows deleted."

➤ Operația 2

Mărirea salariilor cu 15% angajaților angajați acum 5 sau mai mulți ani.

➤ Cod SQL

```
UPDATE ANGAJAT
SET SALARIU = SALARIU*1.15
WHERE ID_ANGAJAT IN (
    SELECT ID_ANGAJAT
    FROM ANGAJAT
    WHERE MONTHS_BETWEEN(SYSDATE, DATA_ANGAJARE) >= 60
);
```

➤ Screen Prints

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

```
UPDATE ANGAJAT
SET SALARIU = SALARIU*1.15
WHERE ID_ANGAJAT IN (
    SELECT ID_ANGAJAT
    FROM ANGAJAT
    WHERE MONTHS_BETWEEN(SYSDATE, DATA_ANGAJARE) >= 60
);
```

In the bottom-right pane, there is a results window titled "Query Result". It displays the message "Task completed in 0.034 seconds" and the output "21 rows updated.".

21 rows updated.

➤ Operația 3

Alterarea specialității inginerilor audio care au un subtip de rock la doar 'Rock'.

➤ Cod SQL

```
UPDATE INGINER_AUDIO
SET SPECIALITATE_GEN = 'Rock'
WHERE ID_INGINER_AUDIO IN (
    SELECT ID_INGINER_AUDIO
    FROM INGINER_AUDIO
    WHERE UPPER(SPECIALITATE_GEN) LIKE '%ROCK%'
);

```

➤ Screen Prints

The screenshot shows a database interface with a script editor and a results viewer. The script editor contains the following SQL code:

```
UPDATE INGINER_AUDIO
SET SPECIALITATE_GEN = 'Rock'
WHERE ID_INGINER_AUDIO IN (
    SELECT ID_INGINER_AUDIO
    FROM INGINER_AUDIO
    WHERE UPPER(SPECIALITATE_GEN) LIKE '%ROCK%'
);

```

The line `FROM INGINER_AUDIO` is highlighted with a yellow background. Below the editor is a toolbar with icons for running, saving, and printing. The results window shows the message "Task completed in 0.038 seconds" and the output "4 rows updated."

---

#### 14. Crearea unei vizualizări complexe

---

**Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.**

Creați un view care să afișeze pentru fiecare trupă id-ul, numele, genul, data înființării, numărul de cântece sub numele acesteia și generația căreia aparține în funcție de vechime astfel: dacă a fost lansată acum mai puțin de 5 ani este Contemporan, peste 5 ani însă mai puțin de 20 de ani este Retro, iar mai vechi este Vintage.

➤ Cod SQL

```
CREATE OR REPLACE VIEW DETALII_TRUPA AS
    SELECT T.ID_TRUPA, T.NUME, T.GEN, T.DATA_INFIINTARE, NR_CANTECE,
           CASE WHEN MONTHS_BETWEEN(SYSDATE, DATA_INFIINTARE) < 60 THEN
    'Contemporan'
           WHEN MONTHS_BETWEEN(SYSDATE, DATA_INFIINTARE) < 240 THEN 'Retro'
           ELSE 'Vintage' END AS GENERATIE
    FROM TRUPA T
    JOIN( SELECT T.ID_TRUPA, COUNT(ID_CANTEC) AS NR_CANTECE
          FROM TRUPA T
          LEFT OUTER JOIN TRUPA_CANTEC TC ON T.ID_TRUPA = TC.ID_TRUPA
          GROUP BY T.ID_TRUPA) NRC ON NRC.ID_TRUPA = T.ID_TRUPA
```

➤ operatie LMD permisa

```
INSERT INTO DETALII_TRUPA(NUME, GEN, DATA_INFIINTARE)
VALUES ('Silent Hill', 'Metal Rock', DATE '1995-02-17')
```

➤ operatie LMD nepermisa

```
INSERT INTO DETALII_TRUPA(NUME, DATA_INFIINTARE)
VALUES ('Vixens', DATE '1935-04-23')
```

deoarece campul 'GEN' din TRUPA nu poate fi null)

➤ Screen Prints

```
--14
--Creați un view care să afișeze pentru fiecare trupă id-ul, numele, genul, data înființării, numărul de cântece
CREATE OR REPLACE VIEW DETALII_TRUPA AS
    SELECT T.ID_TRUPA, T.NUME, T.GEN, T.DATA_INFIINTARE, NR_CANTECE,
           CASE WHEN MONTHS_BETWEEN(SYSDATE, DATA_INFIINTARE) < 60 THEN 'Contemporan'
                 WHEN MONTHS_BETWEEN(SYSDATE, DATA_INFIINTARE) < 240 THEN 'Retro'
                 ELSE 'Vintage' END AS GENERATIE
    FROM TRUPA T
    JOIN( SELECT T.ID_TRUPA, COUNT(ID_CANTEC) AS NR_CANTECE
          FROM TRUPA T
          LEFT OUTER JOIN TRUPA_CANTEC TC ON T.ID_TRUPA = TC.ID_TRUPA
          GROUP BY T.ID_TRUPA) NRC ON NRC.ID_TRUPA = T.ID_TRUPA;
```

Query Result x Script Output x | Task completed in 0.216 seconds

ACTION... View DETALII\_TRUPA created.

```
SELECT * FROM DETALII_TRUPA;
```

Script Output x Query Result x | All Rows Fetched: 7 in 0.075 seconds

ID_TRUPA	NUME	GEN	DATA_INFIINTARE	NR_CANTECE	GENERATIE
1	My Chemical Romance	Alternative rock	18-SEP-01	7	Vintage
2	Gerard Way	Alternative rock	20-DEC-12	1	Retro
3	Frank Iero	Punk rock	09-JUN-14	2	Retro
4	The Future Violents	Alternative rock	13-NOV-18	3	Retro
5	TWRP	Funk rock	05-APR-11	7	Retro
6	Tom Cardy	Musical Comedy	06-JUL-23	1	Contemporan
7	Queen	Progressive Rock	28-FEB-70	6	Vintage

```
--operatie LMD permisa
INSERT INTO DETALII_TRUPA(NUME, GEN, DATA_INFIINTARE)
VALUES ('Silent Hill', 'Metal Rock', DATE '1995-02-17');
```

Script Output x Query Result x | Task completed in 0.076 seconds

1 row inserted.

```
--operatie LMD nepermisa
INSERT INTO DETALII_TRUPA(NUME, DATA_INFIINTARE)
VALUES ('Vixens', DATE '1935-04-23');
```

Script Output x Query Result x | Task completed in 0.062 seconds

ORA-01400: cannot insert NULL into ("LUCA"."TRUPA"."GEN")

---

**15. Formulați în limbaj natural și implementați în SQL cereri cu Outer Join, Division și Analiză-n.**

---

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

➤ Cerere cu Outer Join

Pentru fiecare angajat afișați titlul, format din numele acestuia și tipul său. Dacă nu are tip, va fi numele departamentului. Ordonați după nume.

➤ Cod SQL

```
SELECT A.NUME || ' - ' ||  
NVL(CASE --functia nvl  
      WHEN MT.ID_ANGAJAT IS NOT NULL THEN 'Manager Trupa'  
      WHEN AC.ID_ANGAJAT IS NOT NULL THEN 'Administrator Contracte'  
      WHEN CM.ID_ANGAJAT IS NOT NULL THEN 'Cercetator Marketing'  
      WHEN IA.ID_ANGAJAT IS NOT NULL THEN 'Inginer Audio'  
      END, 'Departament ' || A.DEPARTAMENT) AS TITLTU  
FROM ANGAJAT A  
FULL OUTER JOIN MANAGER_TRUPA MT ON A.ID_ANGAJAT = MT.ID_ANGAJAT  
FULL OUTER JOIN ADMINISTRATOR_CONTRACTE AC ON A.ID_ANGAJAT = AC.ID_ANGAJAT  
FULL OUTER JOIN CERCETATOR_MARKETING CM ON A.ID_ANGAJAT = CM.ID_ANGAJAT  
FULL OUTER JOIN INGINER_AUDIO IA ON A.ID_ANGAJAT = IA.ID_ANGAJAT  
ORDER BY A.NUME;
```

➤ Screen Prints

```
④ SELECT A.NUME || ' - ' ||  
      NVL(CASE --functia nvl  
            WHEN MT.ID_ANGAJAT IS NOT NULL THEN 'Manager Trupa'  
            WHEN AC.ID_ANGAJAT IS NOT NULL THEN 'Administrator Contracte'  
            WHEN CM.ID_ANGAJAT IS NOT NULL THEN 'Cercetator Marketing'  
            WHEN IA.ID_ANGAJAT IS NOT NULL THEN 'Inginer Audio'  
            END, 'Departament ' || A.DEPARTAMENT) AS TITLTU  
FROM ANGAJAT A  
FULL OUTER JOIN MANAGER_TRUPA MT ON A.ID_ANGAJAT = MT.ID_ANGAJAT  
FULL OUTER JOIN ADMINISTRATOR_CONTRACTE AC ON A.ID_ANGAJAT = AC.ID_ANGAJAT  
FULL OUTER JOIN CERCETATOR_MARKETING CM ON A.ID_ANGAJAT = CM.ID_ANGAJAT  
FULL OUTER JOIN INGINER_AUDIO IA ON A.ID_ANGAJAT = IA.ID_ANGAJAT  
ORDER BY A.NUME;
```

Script Output x Query Result x

SQL | All Rows Fetched: 25 in 0.016 seconds

TITLTU
1 Bobby Singer - Departament Productie
2 Brian Schechter - Manager Trupa
3 Bruce Wayne - Departament Financiar
4 Cardinal Copia - Cercetator Marketing
5 Cass Tiel - Inginer Audio
6 Clark Kent - Departament Financiar
7 Cole Wearth - Inginer Audio
8 Dean Winchester - Inginer Audio
9 Dylan Dove - Manager Trupa
10 Geralt Rivia - Administrator Contracte
11 Harley Quinn - Administrator Contracte
12 Jane Doe - Cercetator Marketing
13 Jay Litting - Departament Vanzari
14 Jim Beach - Manager Trupa
15 John Doe - Administrator Contracte
16 Kai Firre - Cercetator Marketing
17 Lola Bandit - Manager Trupa
18 Miguel O'hara - Cercetator Marketing
19 Miles Morales - Inginer Audio
20 Nya Walter - Departament Productie
21 Peter Parker - Inginer Audio
22 Sam Winchester - Administrator Contracte
23 Tim Curry - Cercetator Marketing
24 Tucker Rule - Manager Trupa
25 Zane Icero - Administrator Contracte

➤ Cerere cu Division

Să se afișeze numele tuturor cântecelor care apar în toate albumele lansate în 2024.

➤ Cod SQL

```
SELECT NUME
FROM CANTEC
WHERE ID_CANTEC IN(
    SELECT DISTINCT C.ID_CANTEC
    FROM CANTEC C
    WHERE NOT EXISTS(
        (SELECT ID_ALBUM
        FROM ALBUM
        WHERE EXTRACT(YEAR FROM DATA_LANSARE) = 2024)
        MINUS
        (SELECT A.ID_ALBUM
        FROM ALBUM A
        JOIN CANTEC_ALBUM CA ON A.ID_ALBUM = CA.ID_ALBUM
        AND CA.ID_CANTEC = C.ID_CANTEC)
    ));
);
```

➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL code for finding songs from 2024 across all albums. The bottom pane shows the execution results in the 'Query Result' tab, which contains a single row with the value '1 Online' under the column labeled 'NUME'.

```
--Division
--Sa se afiseze numele tuturor cantecelor care apar in toate albumele lansate in 2024
SELECT NUME
FROM CANTEC
WHERE ID_CANTEC IN(
    SELECT DISTINCT C.ID_CANTEC
    FROM CANTEC C
    WHERE NOT EXISTS(
        (SELECT ID_ALBUM
        FROM ALBUM
        WHERE EXTRACT(YEAR FROM DATA_LANSARE) = 2024)
        MINUS
        (SELECT A.ID_ALBUM
        FROM ALBUM A
        JOIN CANTEC_ALBUM CA ON A.ID_ALBUM = CA.ID_ALBUM
        AND CA.ID_CANTEC = C.ID_CANTEC)
    ));
);
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.039 seconds

NUME
1 Online

➤ Cerere cu analiză top-n

Afişați detalii despre membrii celor mai vechi 2 trupe.

➤ Cod SQL

```
SELECT M.NUME, M.ZI_NASTERE, M.GEN
FROM (
    SELECT M2.NUME, M2.ZI_NASTERE, M2.GEN, MT.ID_TRUPA
    FROM MUZICIAN M2
    JOIN MEMBRU_TRUPA MT ON M2.ID_MUZICIAN = MT.ID_MUZICIAN
) M
JOIN (SELECT ID_TRUPA
      FROM (SELECT ID_TRUPA
            FROM TRUPA
            ORDER BY DATA_INFIINTARE)
      WHERE ROWNUM <= 2) T ON M.ID_TRUPA = T.ID_TRUPA;
```

➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. The top half displays the SQL query in the 'Script' tab, with the 'SQL' icon selected. The bottom half shows the 'Query Result' tab, which displays the output of the query. The output is a table with columns: NUME, ZI\_NASTERE, and GEN. The data rows are numbered 1 through 8, listing the names, birth dates, and genders of the members of the two oldest bands.

	NUME	ZI_NASTERE	GEN
1	Freddy Mercury	05-SEP-46	m
2	Brian May	19-JUL-47	m
3	John Deacon	19-SEP-51	m
4	Roger Taylor	26-JUL-49	m
5	Gerard Way	09-APR-77	x
6	Mikey Way	10-SEP-80	m
7	Ray Toro	15-JUL-77	m
8	Frank Iero	31-OCT-81	m

---

## 16. Optimizarea unei cereri

---

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

➤ Cererea

Afişați detalii despre cântecele care au cel puțin 3 minute și apar pe cel puțin un album de tip 'EP'.

➤ Expresia algebraică inițială:

$\rho_1 = \text{SELECT} (\text{ALBUM}, \text{TIP} = "EP")$

$\rho_2 = \text{JOIN} (\rho_1, \text{CANTEC\_ALBUM})$

$\rho_3 = \text{JOIN} (\text{CANTEC}, \rho_2)$

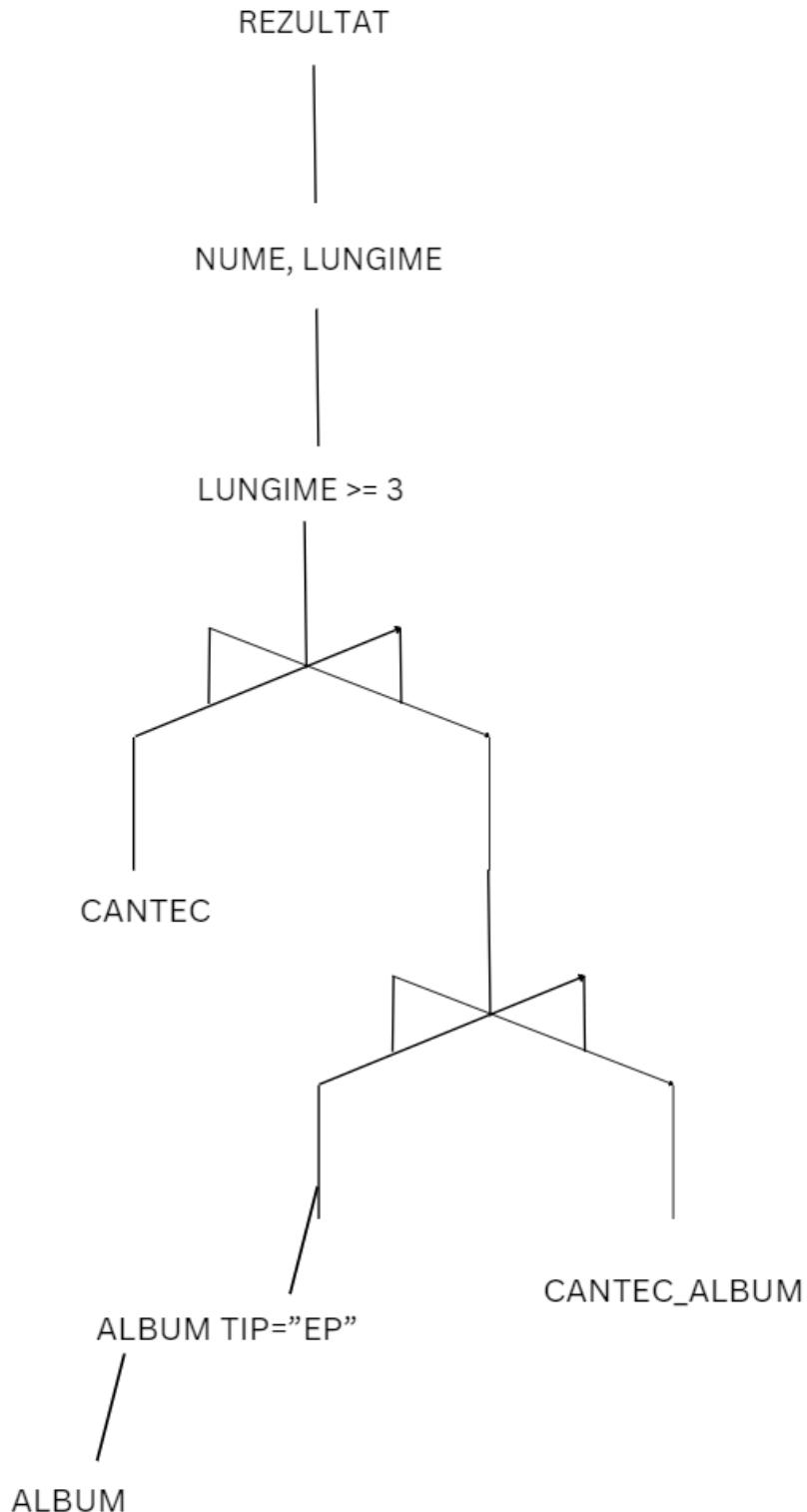
$\rho_4 = \text{SELECT} (\rho_3, \text{lungime} \geq 3)$

$\rho_{\text{final}} = \text{PROJECT} (\text{nume}, \text{lungime})$

➤ Cod SQL

```
SELECT C.NUME, C.LUNGIME
FROM CANTEC C
JOIN (
  SELECT CA.ID_CANTEC
  FROM CANTEC_ALBUM CA
  JOIN (
    SELECT ID_ALBUM
    FROM ALBUM
    WHERE TIP = 'EP'
  ) A ON CA.ID_ALBUM = A.ID_ALBUM
) X ON C.ID_CANTEC = X.ID_CANTEC
WHERE C.LUNGIME >= 3;
```

➤ Arborele algebric inițial:



➤ Optimizare SQL

Comutativitatea și asociativitatea înlățuirilor: ordinea înlățuirilor poate fi reordonată pentru a reduce costul de execuție. Selectia pe lungime  $\geq 3$  minute poate fi mutată înainte de înlățuirea cu CANTEC\_ALBUM pentru a reduce volumul de date.

➤ Expresie algebraică optimizată

$\rho_1 = \text{JOIN}(\text{CANTEC}, \text{CANTEC\_ALBUM})$

$\rho_2 = \text{SELECT}(\text{ALBUM}, \text{TIP}=\text{"EP"})$

$\rho_3 = \text{JOIN}(\rho_1, \rho_2)$

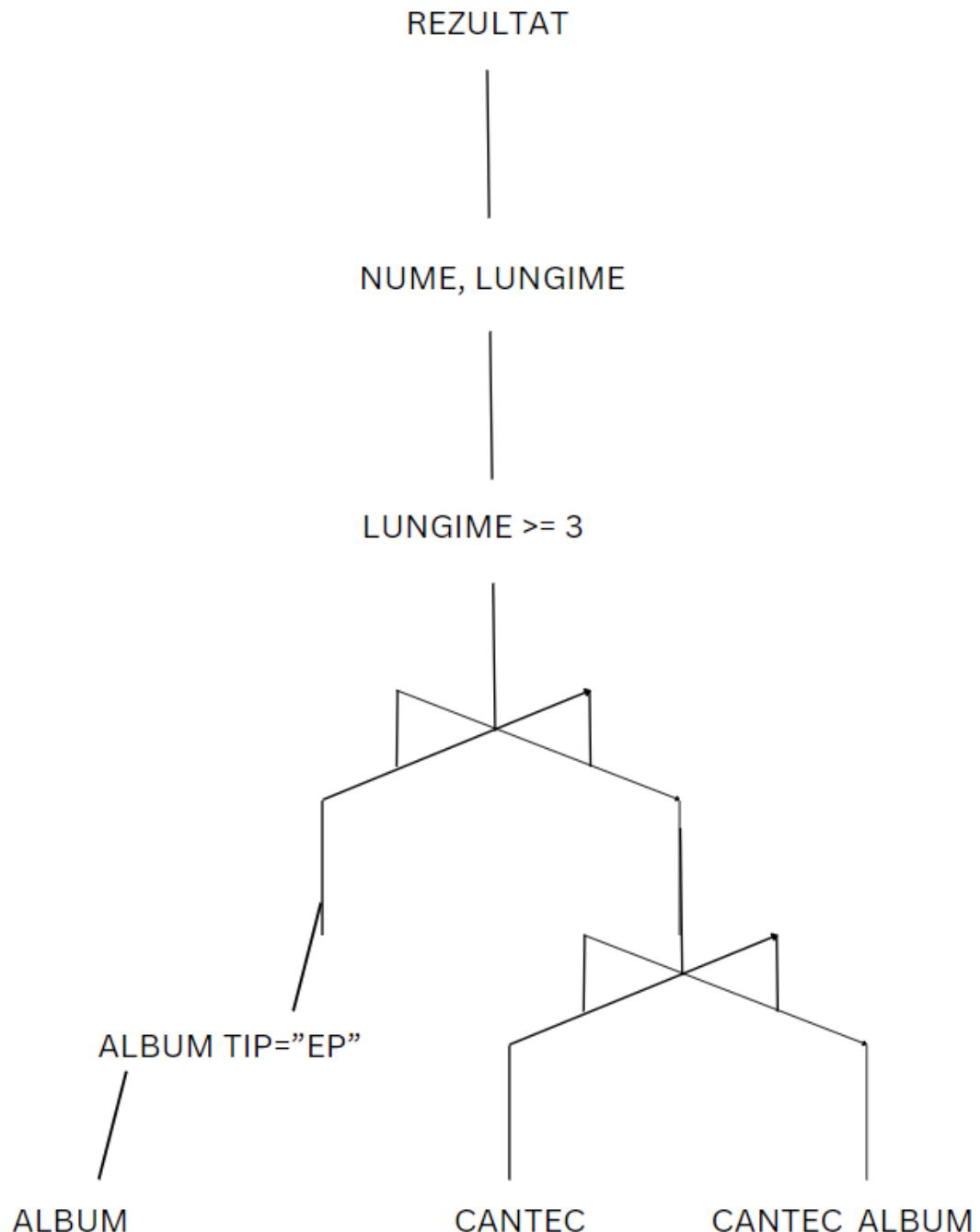
$\rho_4 = \text{SELECT}(\rho_3, \text{lungime} \geq 3)$

$\rho_{\text{final}} = \text{PROJECT}(\rho_4, \text{nume}, \text{lungime})$

➤ Cod SQL

```
SELECT C.NUME, C.LUNGIME
FROM CANTEC C
JOIN CANTEC_ALBUM CA ON C.ID_CANTEC = CA.ID_CANTEC
JOIN (
    SELECT ID_ALBUM
    FROM ALBUM
    WHERE TIP = 'EP'
) A ON CA.ID_ALBUM = A.ID_ALBUM
WHERE C.LUNGIME >= 3;
```

- Arboarele algebraic optimizat



---

**17. Realizarea normalizării BCNF, FN4, FN5. Aplicarea denormalizării.**

---

a. Realizarea normalizării BCNF, FN4, FN5.

➤ BCNF

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

Pentru a demonstra această normalizare voi folosi tabela ALBUM. Pentru tabela de mai jos știm că atributul ID\_ALBUM este cheia primară. În tabelul non BCNF avem doar attribute prime.

ID_ALBUM	ID_CASA_DISCURI	ID_TRUPA	TITLU	TIP	DATA_LANSARII
1	1	1	„Helena – Single”	Single	2004-04-16
4	3	4	„Barriers - EP”	EP	2004-06-08
8	5	7	„Best Hits – Album”	Album	2001-03-13

Putem observa că atributul TIP este dependent de atributul TITLU. Prin urmare tabela noastră nu este în BCNF. Pentru a schimba acest lucru transformăm TITLU în NUME pentru că atributul NUME să nu mai depindă de acesta.

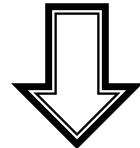
ID_ALBUM	ID_CASA_DISCURI	ID_TRUPA	NUME	TIP	DATA_LANSARII
1	1	1	„Helena”	Single	2004-04-16
4	3	4	„Barriers”	EP	2004-06-08
8	5	7	„Best Hits”	Album	2001-03-13

➤ FN4

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.

Pentru a demonstra această normalizare voi folosi tabela ALBUM\_TRUPA\_CANTECE în care avem dependențe ID\_CANTEC și ID\_TRUPA cu ID\_CANTEC și ID\_ALBUM. Tabelul este în BCNF deci rămâne să descompunem în 2 tabele:

ID_CANTEC	ID_TRUPA	ID_ALBUM
-----------	----------	----------



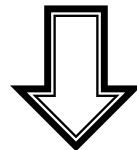
CANTEC_ALBUM	
ID_ALBUM	ID_CANTEC

TRUPA_CANTEC	
ID_TRUPA	ID_CANTEC

➤ FN5

Un tabel se află în FN5 dacă acesta se află în FN4 și elimină redundanțele datorate dependenței ciclice. Se suprimă toate join-dependențele care nu sunt implicate de o cheie. Ca exemplu vom lua același tabel de la FN4. Avem drept cheie primară toate cele 3 ID-uri doar că tabelul poate fi spart în 2 fără ca informațiile să fie pierdute. Acesta nu se află în FN5. Spargem tabelul în alte 2 tabele care printr-un join duc la tabela inițială fără a se pierde informație sau să adăugați redundanță.

ID_CANTEC	ID_TRUPA	ID_ALBUM
-----------	----------	----------



CANTEC_ALBUM	
ID_ALBUM	ID_CANTEC

TRUPA_CANTEC	
ID_TRUPA	ID_CANTEC

b. Aplicarea denormalizării, justificând necesitatea acesteia.

Denormalizarea constă în adăugare de date redundante pentru a crește viteza interogărilor la baza de date. De exemplu în proiectul realizat putem adăugala o coloană pentru numele trupei care a scris albumul pentru a evita join-ul repetat între tabelele ALBUM și TRUPA. Pentru a spori eficiența denormalizării aceasta trebuie realizată după ce sunt studiate tipurile de query-uri care se fac pe baza de date.