

# Proiect Sisteme de Gestiune a Bazelor de Date Gestionarea unor case de discuri

Proiect realizat de Șerba Ioana-Raluca  
Seria 24 Grupa 241 Anul Universitar 2024-2025

## Cuprins

.....	1
<b>1. Prezentați pe scurt baza de date (utilitatea ei) .....</b>	<b>5</b>
<b>2. Realizați diagrama entitate-relație (ERD):.....</b>	<b>7</b>
<b>3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus.....</b>	<b>8</b>
<b>4. Implementați în Oracle diagrama conceptuală realizată.....</b>	<b>9</b>
1. Tabela CASA_DISCURI .....	10
2. Tabela MUZICIAN.....	11
3. Tabela CONTRACT .....	12
4. Tabela TRUPA .....	13
5. Tabela MEMBRU_TRUPA .....	14
6. Tabela INSTRUMENT .....	15
7. Tabela CANTEC .....	16
8. Tabela TRUPA_CANTEC .....	17
9. Tabela ALBUM .....	18
10. Tabela CANTEC_ALBUM .....	19
11. Tabela ANGAJAT .....	20
12. Tabela MANAGER_TRUPA .....	21

13.	Tabela ADMINISTRATOR_CONTRACTE .....	22
14.	Tabela CERCETATOR_MARKETING.....	23
15.	Tabela INGINER_AUDIO .....	24
<b>5.</b>	<b>Adăugați informații coerente în tabelele create.....</b>	<b>25</b>
1.	Tabela CASA_DISCURI .....	25
2.	Tabela MUZICIAN.....	27
3.	Tabela CONTRACT .....	30
4.	Tabela TRUPA .....	33
5.	Tabela MEMBRU_TRUPA .....	35
6.	Tabela INSTRUMENT .....	38
7.	Tabela CANTEC .....	41
8.	Tabela TRUPA_CANTEC .....	46
9.	Tabela ALBUM .....	51
10.	Tabela CANTEC_ALBUM .....	52
11.	Tabela ANGAJAT .....	58
12.	Tabela MANAGER_TRUPA .....	62
13.	Tabela ADMINISTRATOR_CONTRACTE .....	63
14.	Tabela CERCETATOR_MARKETING.....	64
15.	Tabela INGINER_AUDIO .....	65

<b>6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate.</b> .....	69
<b>7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.</b> .....	73
<b>8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție.</b> .....	76
<b>9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură.</b> .....	82
<b>10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.</b> ..	89
<b>11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.</b> .....	92
<b>12. Definiți un trigger de tip LDD. Declanșați trigger-ul.</b> .....	96
<b>13. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet</b> .....	98

---

## 1. Prezentați pe scurt baza de date (utilitatea ei)

---

Baza de date și tema acesutii proiect 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 și stochează informațiile unei case de discuri. O casă de discuri poate să aibă contracte cu muzicieni și poate să dețină albume. Albumele apar sub numele 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 uneia 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. 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.

Funcționalitatea acestei baze 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 deține mai multe case de discuri.

**Infrastructura utilizată pentru implementarea proiectului:**

Versiunea utilizată:

```
BANNER
-----
BANNER_FULL
-----
BANNER_LEGACY
-----
CON_ID
-----
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
          0

BANNER
-----
BANNER_FULL
-----
BANNER_LEGACY
-----
CON_ID
-----
```

Configurația hardware - RAM alocat:

```
Total System Global Area 1610610072 bytes
Fixed Size                  9856408 bytes
Variable Size                905969664 bytes
Database Buffers             687865856 bytes
Redo Buffers                 6918144 bytes
Database mounted.
Database opened.
```

Configurația software - sistem de operare:

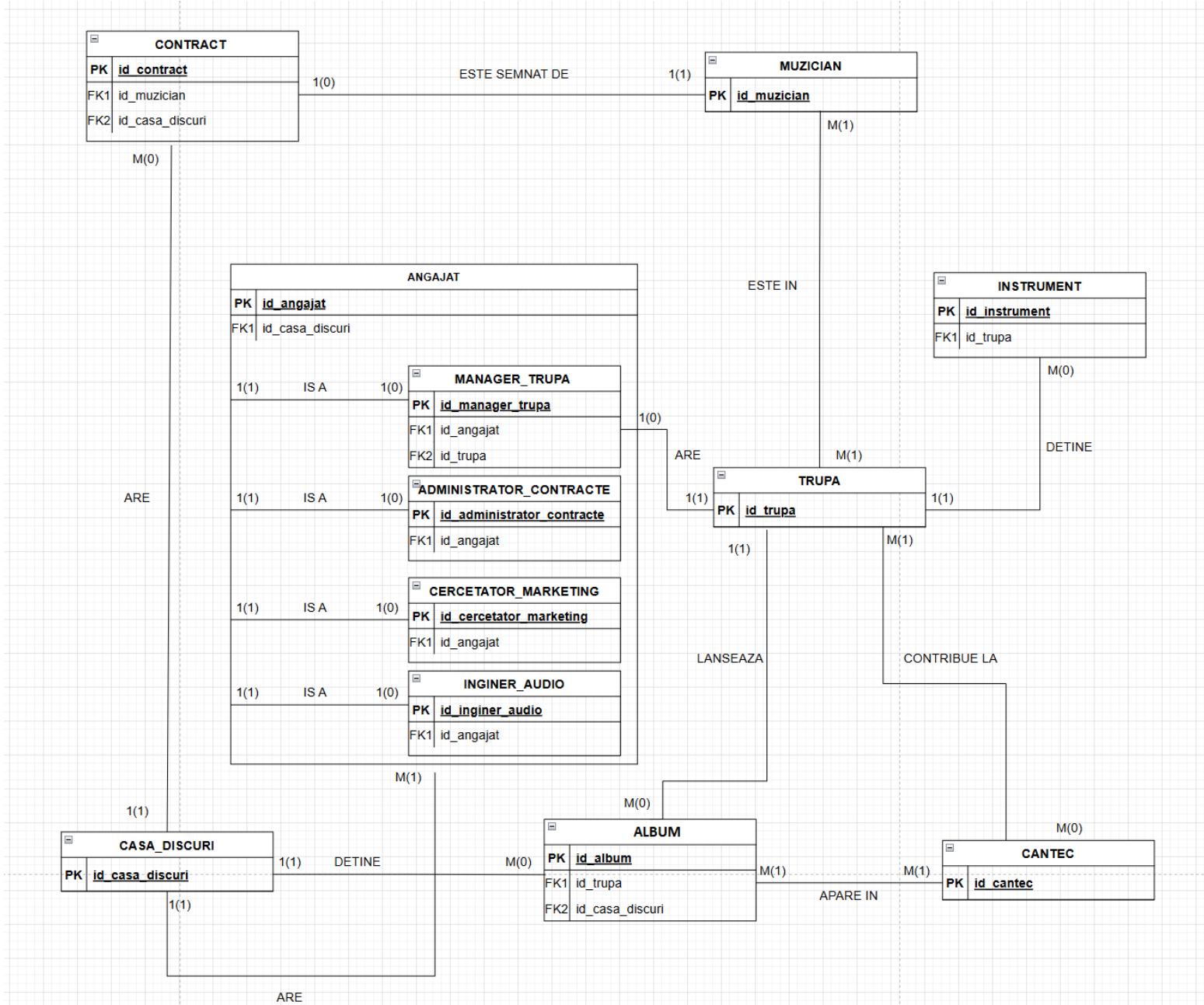
## Windows specifications

Edition                    Windows 10 Pro

Utilizarea sau nu a unei mașini virtuale: Nu.

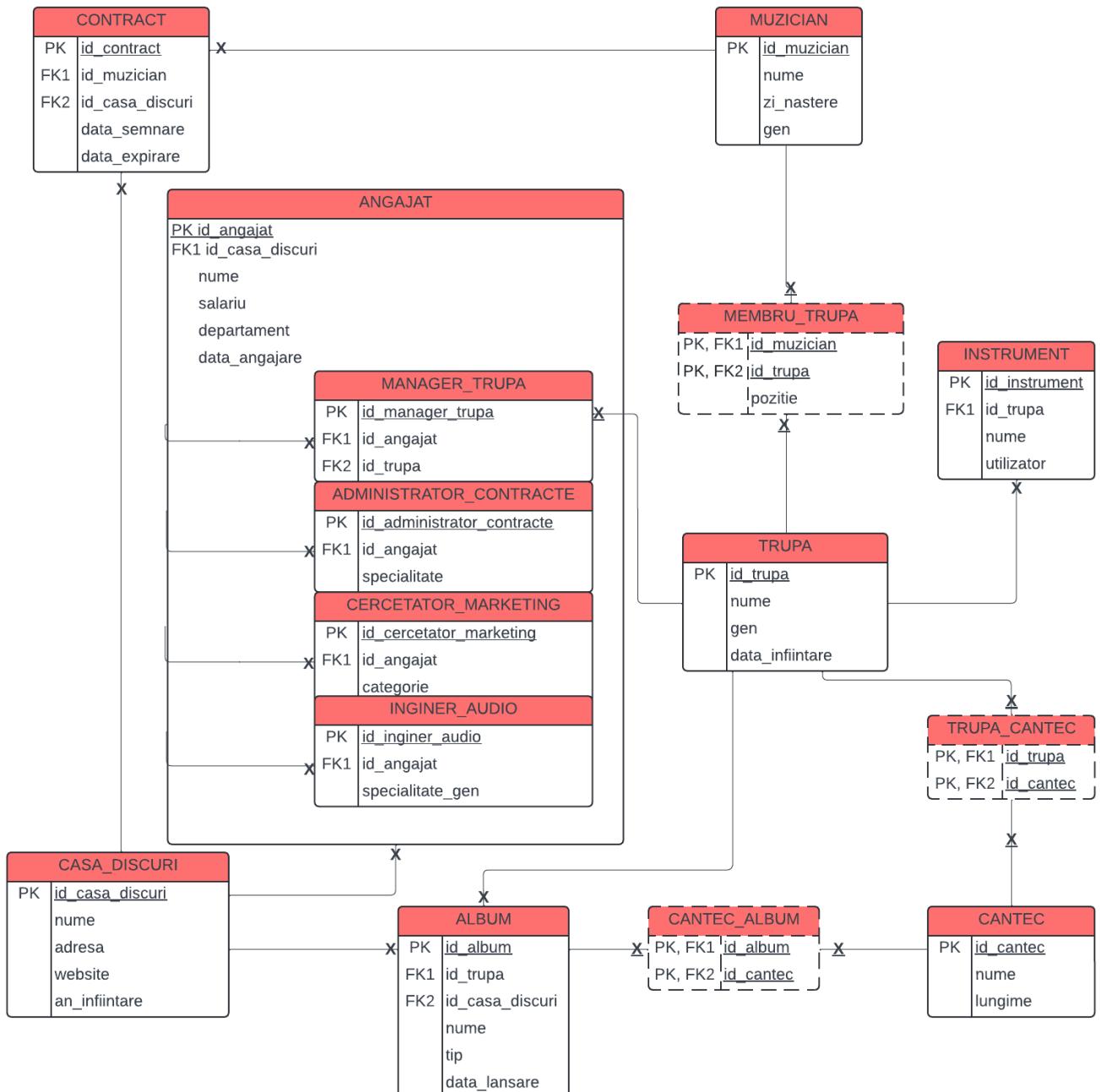
## 2. Realizați diagrama entitate-relație (ERD):

entitățile, relațiile și atributele trebuie definite în **limba română** (vezi curs SGBD, model de diagrama entitate-relație; nu se va accepta alt format).



**3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus.**

înțelegând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română



---

#### 4. Implementați în Oracle diagrama conceptuală realizată

---

definiți toate tabelele, adăugând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

➤ Cod Sursă:

```
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;
```

## 1. Tabela CASA\_DISCURI

### ➤ Cod SQL

```
--casa discuri plsql
DECLARE
    AN_CURENT varchar2(4);
BEGIN
    AN_CURENT := TO_CHAR(SYSDATE, 'YYYY');
    execute immediate '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_AN2 CHECK (AN_INFIINTARE >= 1889 AND AN_INFIINTARE
        <=' || AN_CURENT || ')';
END;
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the PL/SQL code provided above. In the bottom-right pane, there is a log window titled "Script Output" which displays the message "PL/SQL procedure successfully completed." Below the log window, there is a status bar indicating "Task completed in 0.041 seconds".

The screenshot shows the Oracle Database Designer interface. It displays the structure of the "CASA\_DISCURI" table. The table has five columns: "ID\_CASA\_DISCURI" (NUMBER), "NUME" (VARCHAR2(40)), "ADRESA" (VARCHAR2(255)), "WEBSITE" (VARCHAR2(255)), and "AN\_INFIINTARE" (NUMBER). The "ID\_CASA\_DISCURI" column is defined as the primary key with a default value from the sequence "CASA\_DISCURI\_SEQ.NEXTVAL". The "NUME" column is unique and not nullable.

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 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)
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the MUZICIAN table. The code is identical to the one shown above. Below the code editor is a toolbar with various icons. To the right of the toolbar is a status bar displaying the message "Task completed in 0.04 seconds". At the bottom of the interface, there is a message window titled "Script Output" which contains the text "Table MUZICIAN created.".

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

Script Output | Task completed in 0.04 seconds

Table MUZICIAN created.

### 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)  
);
```

➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the CONTRACT table. The code is identical to the one shown in the previous text block. In the bottom-right pane, there is a 'Script Output' window. It displays the message 'Table CONTRACT created.' and indicates that the task completed in 0.043 seconds. Below the output window are several small icons: a red circle with a white exclamation mark, a pencil, a blue square, a white square, and a green triangle.

```
--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)  
);
```

Script Output x  
| Task completed in 0.043 seconds

Table CONTRACT created.

#### 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
);
```

➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the TRUPA table. The code is identical to the one shown above. The word 'CREATE' is highlighted in blue. The bottom-right pane is titled 'Script Output' and displays the message 'Table TRUPA created.' followed by a note that the task completed in 0.037 seconds. There are also several small icons in the bottom right corner of the output pane.

```
--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
);
```

Script Output | Task completed in 0.037 seconds

Table TRUPA created.

## 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'))
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the MEMBRU\_TRUPA table. The code is identical to the one provided above. In the bottom-right pane, there is a 'Script Output' window with a yellow header bar. The output shows the message 'Table MEMBRU\_TRUPA created.' and a note indicating the task completed in 0.045 seconds. The interface includes standard toolbars and a status bar at the bottom.

```
--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'))
);
```

Script Output x

| Task completed in 0.045 seconds

Table MEMBRU\_TRUPA created.

## 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'))
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the INSTRUMENT table. The code is identical to the one shown in the previous text block. In the bottom-right pane, there is a 'Script Output' window with a toolbar. The output window displays the message 'Table INSTRUMENT created.' and indicates that the task completed in 0.049 seconds.

```
--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'))
);
```

Script Output | Task completed in 0.049 seconds

Table INSTRUMENT created.

## 7. Tabela CANTEC

### ➤ Cod SQL

```
--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 )
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a tree view with a node labeled "cantec". Underneath it, the SQL script for creating the table is displayed:

```
--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 )
);
```

In the bottom-right pane, there is a "Script Output" window. It contains the message "Table CANTEC created." and a note indicating the task completed in 0.049 seconds.

## 8. Tabela TRUPA\_CANTEC

### ➤ Cod SQL

```
--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)
);
```

### ➤ Screen Prints

The screenshot shows a database interface with a script editor and an output window.

In the script editor, the SQL code for creating the TRUPA\_CANTEC table is displayed. The code includes a primary key constraint and foreign key references to the TRUPA and CANTEC tables.

```
--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)
);
```

In the output window, the message "Table TRUPA\_CANTEC created." is visible, indicating the successful execution of the command.

## 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') )  
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the ALBUM table. The code is identical to the one provided above. The bottom-left pane is titled "Script Output" and displays the message "Table ALBUM created." followed by a timestamp "Task completed in 0.051 seconds". The bottom right pane shows some standard icons.

```
--  
--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') )  
);
```

Script Output x

Table ALBUM created.  
Task completed in 0.051 seconds

## 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)
);
```

### ➤ Screen Prints

The screenshot shows a database interface with a script editor and a script output window. The script editor contains the SQL code for creating the CANTEC\_ALBUM table, which includes columns for ID\_ALBUM and ID\_CANTEC, a primary key constraint, and foreign key references to the CANTEC and ALBUM tables. The script output window below shows the message "Table CANTEC\_ALBUM created." and indicates the task completed in 0.051 seconds.

```
--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)
);
```

Script Output | Task completed in 0.051 seconds

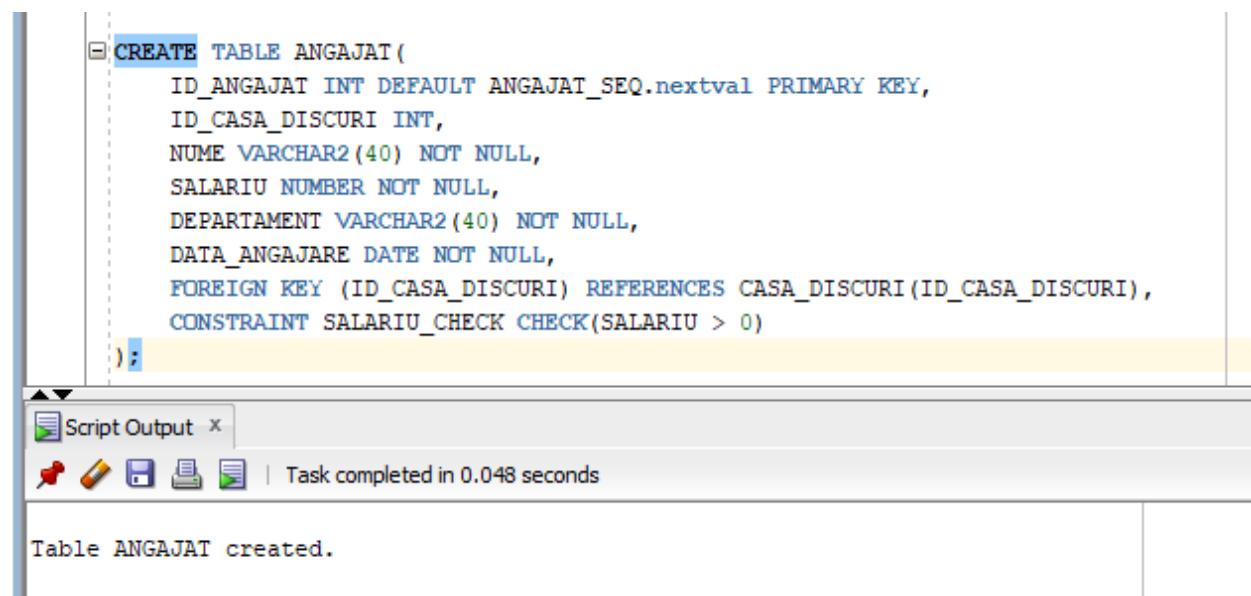
Table CANTEC\_ALBUM created.

## 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)  
);
```

### ➤ Screen Prints



The screenshot shows the Oracle SQL Developer interface. In the main editor area, the SQL code for creating the ANGAJAT table is displayed. The code defines the table structure with columns for ID\_ANGAJAT (primary key), ID\_CASA\_DISCURI, NUME, SALARIU, DEPARTAMENT, and DATA\_ANGAJARE. It includes a foreign key constraint linking to the CASA\_DISCURI table and a check constraint ensuring SALARIU is greater than 0. The SQL statement ends with a closing parenthesis and a semicolon. Below the editor, the 'Script Output' tab is selected, showing the message 'Table ANGAJAT created.' and a note indicating the task completed in 0.048 seconds.

```
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)  
);
```

Script Output | Task completed in 0.048 seconds

Table ANGAJAT created.

## 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)  
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the MANAGER\_TRUPA table. The code is identical to the one shown above. In the bottom-right pane, there is a 'Script Output' window with a toolbar. The output window displays the message 'Table MANAGER\_TRUPA created.' and indicates that the task completed in 0.053 seconds.

```
--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)  
);
```

Script Output x

Table MANAGER\_TRUPA created.  
Task completed in 0.053 seconds

### 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)  
);
```

#### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the table. The code is identical to the one shown above. In the bottom-right pane, there is a 'Script Output' window. It displays the message 'Table ADMINISTRATOR\_CONTRACTE created.' and indicates that the task completed in 0.048 seconds. The interface includes standard toolbars and a status bar at the bottom.

```
--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)  
);
```

Script Output | Task completed in 0.048 seconds

Table ADMINISTRATOR\_CONTRACTE created.

## 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)  
);
```

### ➤ Screen Prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the CERCETATOR\_MARKETING table. The code is identical to the one shown in the previous text block. In the bottom-right pane, there is a 'Script Output' window showing the result of the execution. The output window has a title bar 'Script Output X' and a message bar 'Task completed in 0.046 seconds'. Below the message bar, the text 'Table CERCETATOR\_MARKETING created.' is displayed.

## 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)
);
```

### ➤ Screen prints

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script for creating the INGINER\_AUDIO table. The code is identical to the one provided above. In the bottom-right pane, there is a 'Script Output' window with a toolbar. The output area displays the message 'Table INGINER\_AUDIO created.' and indicates that the task completed in 0.048 seconds.

```
--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)
);
```

Script Output x

Table INGINER\_AUDIO created.

Task completed in 0.048 seconds

---

## 5. Adăugați informații coerente în tabelele create

---

(minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru fiecare tabelă asociativă).

### 1. Tabela CASA\_DISCURI

#### ➤ Cod SQL

```
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

```
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

```
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

```
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');
```

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

```
SELECT *
FROM MUZICIAN;
```

In the bottom-right pane, there is a "Query Result" window titled "les". It displays the results of the query. The results are presented in a table with the following columns:

	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

The results show 15 rows of data from the MUZICIAN table. The columns are labeled ID\_MUZICIAN, NUME, ZI\_NASTERE, and GEN. The data includes names like Gerard Way, Mikey Way, Ray Toro, Frank Iero, Evan Nestor, Kayleigh Goldsworthy, Doctor Sung, Commander Meouch, Lord Phobos, Haave Hogan, Tom Cardy, Freddy Mercury, Brian May, John Deacon, and Roger Taylor, along with their birth dates and gender (x for male, m for female, or f for female). Some fields like ZI\_NASTERE and GEN contain null values.

### 3. Tabela CONTRACT

#### ➤ Cod SQL

```
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

```
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

```
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

```
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 MySQL Workbench environment. In the top-left pane, there is a code editor window containing the SQL query:

```
SELECT *
FROM TRUPA;
```

In the bottom-right pane, there is a "Query Result" grid displaying the data from the TRUPA table:

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

## 5. Tabela MEMBRU\_TRUPA

### ➤ Cod SQL

```
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

```
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

```
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

Worksheet    Query Builder

```
--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

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

```
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 ('HTMLLOVE',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);
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
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

```
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

```
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);
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
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

```
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);
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
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

```
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

```
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;

The screenshot shows a MySQL Workbench interface with a query result window titled "Query Result". The SQL query "SELECT \* FROM ALBUM;" has been run, and the results are displayed in a table. The table has columns: ID\_ALBUM, ID\_TRUPA, ID\_CASA\_DISCURI, NUME, TIP, and DATA\_LANSARE. The data consists of 8 rows, each representing an album entry with its details.

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

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

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

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
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);
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
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

```
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);
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
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

```
--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');
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
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

```
--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

```
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

```
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);
```

The screenshot shows a database interface with a query window containing the following SQL statement:

```
SELECT * FROM MANAGER_TRUPA;
```

The results are displayed in a table with four columns: ID\_MANAGER\_TRUPA, ID\_TRUPA, ID\_ANGAJAT. The data is as follows:

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

### 13. Tabela ADMINISTRATOR\_CONTRACTE

#### ➤ Cod SQL

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

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

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

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

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

#### ➤ Screen Prints

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

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

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

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

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

SELECT \* FROM ADMINISTRATOR\_CONTRACTE;

Script Output x Query Result x

P S Q All Rows Fetched: 5 in 0.184 seconds

ID_ADMINISTRATOR_CONTRACTE	ID_ANGAJAT	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

```
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

```
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

```
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

```
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');

SELECT * FROM INGINER_AUDIO;
```

The screenshot shows the MySQL Workbench interface. The top part displays the SQL code for inserting data into the INGINER\_AUDIO table. The bottom part shows the 'Query Result' tab with the output of the 'SELECT \* FROM INGINER\_AUDIO;' query, which returns five rows of data:

ID_INGINER_AUDIO	ID_ANGAJAT	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

Serba Ioana-Raluca  
Seria 24 Grupa 241

--Date inserate pentru exercitiul 8

--ca sa exemplific si exceptia de NO\_CONTRACT, o sa inserez o noua casa de discuri fara  
contracte

```
INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)  
VALUES ('Angel Records', 'fallenangel.com', 2014);
```

--Date inserate pentru exercitiul 9

```
INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)  
VALUES ('RedhoodRecords', 'https://www.rhrecords.com/', 2007); --41  
  
INSERT INTO CASA_DISCURI (NUME, WEBSITE, AN_INFIINTARE)  
VALUES ('Shadow Records', 'https://www.possesme.com/', 2012); --42  
  
INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)  
VALUES ('Will Wood', 'Avant-pop', DATE'2020-04-01'); --21  
  
INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)  
VALUES ('The Tapeworms', 'Chaotic Jazz', DATE'2015-11-06'); --23  
  
INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)  
VALUES ('Ashnikko', 'Hyperpop', DATE'2012-10-31'); --24
```

--WW  
INSERT INTO ALBUM(ID\_TRUPA, ID\_CASA\_DISCURI, NUME, TIP, DATA\_LANSARE)  
VALUES(21, 41, 'The Normal Album', 'EP', DATE '2020-06-10'); --41  
  
--TT  
INSERT INTO ALBUM(ID\_TRUPA, ID\_CASA\_DISCURI, NUME, TIP, DATA\_LANSARE)  
VALUES(23, 41, 'Everything is a lot', 'EP', DATE '2015-05-20'); --42

--ASHN  
INSERT INTO ALBUM(ID\_TRUPA, ID\_CASA\_DISCURI, NUME, TIP, DATA\_LANSARE)  
VALUES(24, 42, 'WEEDKILLER', 'Album', DATE '2023-08-25'); --43

--WW  
INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Suburbia Overture', 6.16); --41

INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('Laplace ''s Angel', 4.01); --42

INSERT INTO CANTEC(NUME, LUNGIME)  
VALUES ('2econd-2ight-2eer', 3.26); --43

INSERT INTO CANTEC\_ALBUM(ID\_ALBUM, ID\_CANTEC)  
VALUES(41, 41);

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(41,42);
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(41,43);

SELECT * FROM CANTEC;

--TT

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Front Street',5.15);--44

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Chemical Overreaction/Everything is a lot',8.28);--45

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(42,44);
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(42,45);

--ASHN

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Cheerleader',2.40);--46

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Chokehold Cherry Python',2.39);--47

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Worms',3.26);--48

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Want It All',2.39);--49

INSERT INTO CANTEC(NUME, LUNGIME)
VALUES ('Dying Star',4.15);--50

INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(43,46);
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(43,47);
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(43,48);
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(43,49);
INSERT INTO CANTEC_ALBUM(ID_ALBUM, ID_CANTEC)
VALUES(43,50);
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

--date introduse pentru exercitiul 13

```
INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('Montaigne', 'NuPop', DATE '2012-04-20'); --41

INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Montaigne Cerro', DATE '1995-08-14', 'f'); --41

INSERT INTO MEMBRU_TRUPA (ID_MUZICIAN, ID_TRUPA, POZITIE)
VALUES (41, 41, 'Vocalist');

INSERT INTO TRUPA_CANTEC (ID_TRUPA, ID_CANTEC)
VALUES (41, 15);

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,
DATA_ANGAJARE)
VALUES (3, 'Pamela Isley', 8700, 'Financiar', DATE '2001-02-28');

INSERT INTO ALBUM (ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES (6, 4, 'Red Flags', 'Single', DATE '2023-03-12');

INSERT INTO ALBUM (ID_TRUPA, ID_CASA_DISCURI, NUME, TIP, DATA_LANSARE)
VALUES (6, 4, 'Red Flags', 'EP', DATE '2024-03-12');

SELECT * FROM trupa;

INSERT INTO CANTEC (NUME, LUNGIME)
VALUES ('The No Pants Dance', 4.31); --61

INSERT INTO CANTEC (NUME, LUNGIME)
VALUES ('The No Pants Dance', 4.28); --62

INSERT INTO TRUPA (NUME, GEN, DATA_INFIINTARE)
VALUES ('NSP', 'Comedy Rock', DATE '2010-04-20'); --42

INSERT INTO TRUPA_CANTEC (ID_TRUPA, ID_CANTEC)
VALUES (5, 61);

INSERT INTO TRUPA_CANTEC (ID_TRUPA, ID_CANTEC)
VALUES (5, 62);

INSERT INTO TRUPA_CANTEC (ID_TRUPA, ID_CANTEC)
VALUES (42, 62);
```

--date au fost introduce pentru a exemplifica exceptiile si cazurile tratate

---

**6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate.**

---

Apelați subprogramul.

➤ **Cerință:**

Pentru o casa de discuri data, sa se afiseze numarul de albume pe care casa le are, memorand albumele intr-un vector. Apoi, afisati cel mai vechi album folosind vectorul.

Folosind un tablou imbricat, retineti si afisati numele tuturor membrilor trupei albumului respectiv, si pentru fiecare pozitie din trupa inafara de vocalist, afisati cate un instrument folosind un tablou indexat (indexat dupa pozitia utilizatorului: basist, vocalist, pianist etc). Afisati daca exista instrumente pentru pozitii neocupate, pozitia si un instrument de acel tip.

➤ Cod sursă:

```
/  
CREATE OR REPLACE PROCEDURE EX6 ( V_CD CASA_DISCURI.ID_CASA_DISCURI%TYPE)  
AS --id casa discuri pt care afisam rezultatele  
  
    TYPE VECTOR_ALBUME IS VARRAY(100) OF ALBUM%ROWTYPE;  
    V_ALBUME VECTOR_ALBUME := VECTOR_ALBUME(); --aici pun albumele - vector  
  
    TYPE DETALII_MUZICIAN IS RECORD (  
        NUME MUZICIAN.NUME%TYPE,  
        VARSTA NUMBER(3),  
        POZITIE MEMBRU_TRUPA.POZITIE%TYPE  
    );  
  
    TYPE TABLOU_MEMBRII IS TABLE OF DETALII_MUZICIAN;  
    T_MEMBRII TABLOU_MEMBRII := TABLOU_MEMBRII(); --TABLOU IMBRICAT CU  
    MUZICIENII  
  
    TYPE TABLOU_INSTRUMENTE IS TABLE OF INSTRUMENT.NUME%TYPE INDEX BY  
    INSTRUMENTUTILIZATOR%TYPE;  
    T_INSTRUMENTE TABLOU_INSTRUMENTE; --TABLOU CU INSTRUMENTE INDEXATE PER  
    POZITIE  
  
    VCOUNT NUMBER(3) := 1;  
    VTRUPA TRUPA.ID_TRUPA%TYPE;  
    INDX INSTRUMENTUTILIZATOR%TYPE;  
  
BEGIN
```

```
--pun albumele unei case de discuri specifice intr-un vector, in
ordinea datii lansarii
FOR I IN (SELECT * FROM ALBUM WHERE ID_CASA_DISCURI = V_CD ORDER BY
DATA_LANSARE DESC) LOOP
    V_ALBUME.EXTEND;
    V_ALBUME(VCOUNT) := I;
    VCOUNT := VCOUNT + 1;
END LOOP;
VCOUNT := VCOUNT - 1;

DBMS_OUTPUT.PUT_LINE('Casa de discuri are ' || V_ALBUME.COUNT || '
albume');

--pentru cel mai vechi album, gasesc id-ul trupei
SELECT ID_TRUPA
INTO VTRUPA
FROM ALBUM
WHERE ID_ALBUM = V_ALBUME(VCOUNT).ID_ALBUM;
--cum e ordonat, cea mai veche e mereu ultima (ca sa folosesc last mai
jos si sa aiba sens)

--CEL mai vechi album,
DBMS_OUTPUT.PUT_LINE('Cel mai vechi album: ' ||
V_ALBUME(V_ALBUME.LAST).NUME );

DBMS_OUTPUT.NEW_LINE;
--tablou imbricat cu membrii
SELECT M.NUME, FLOOR(MONTHS_BETWEEN(SYSDATE,M.ZI_NASTERE)/12) AS
VARSTA,
        MT.POZITIE
BULK COLLECT INTO T_MEMBRII
FROM TRUPA T
JOIN MEMBRU_TRUPA MT ON T.ID_TRUPA = MT.ID_TRUPA
JOIN MUZICIAN M ON M.ID_MUZICIAN = MT.ID_MUZICIAN
WHERE T.ID_TRUPA = VTRUPA;

--afisarea membrilor trupei
FOR I IN T_MEMBRII.FIRST..T_MEMBRII.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(T_MEMBRII(I).NUME);
END LOOP;

--tabel indexat cu UTILIZATORUL cu instrumentele TRUPEI
FOR I IN (SELECT NUME, UTILIZATOR FROM INSTRUMENT WHERE ID_TRUPA =
VTRUPA) LOOP
    T_INSTRUMENTE(I.UTILIZATOR) := I.NUME;
END LOOP;
--se suprascriu cele care repeta utilizatorul - ramane 1

DBMS_OUTPUT.NEW_LINE;
--cate un instrument din fiecare pozitie care apare in trupa
--in afara de vocalist (nu are instrumente)
FOR I IN (SELECT UNIQUE POZITIE FROM MEMBRU_TRUPA WHERE ID_TRUPA =
VTRUPA AND POZITIE != 'Vocalist') LOOP
    DBMS_OUTPUT.PUT_LINE(I.POZITIE || ':' || |
T_INSTRUMENTE(I.POZITIE));
    T_INSTRUMENTE.DELETE(I.POZITIE);
END LOOP;
```

```

--daca mai sunt instrumente
IF T_INSTRUMENTE.COUNT > 0 THEN
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE('Exista instrumente pentru pozitii neocupate:');
END;

--iteram prin ele pana nu mai am index
INDX := T_INSTRUMENTE.FIRST;
WHILE INDX IS NOT NULL LOOP
    DBMS_OUTPUT.PUT_LINE(INDX || ':' || T_INSTRUMENTE(INDX));
    INDX := T_INSTRUMENTE.NEXT(INDX);
END LOOP;
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20000, 'nu au fost
gasite date');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001, 'au fost
returnate prea multe linii');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002, 'alta eroare');

END;
/
-----/
BEGIN
    EX6(5);
END;

```

## ➤ Printscreen-uri

```

CREATE OR REPLACE PROCEDURE EX6 ( V_CD CASA_DISCURI.ID_CASA_DISCURI%TYPE) AS --id casa discuri pt care afisam rezultatele
TYPE VECTOR_ALBUM IS VARRAY(100) OF ALBUM%ROWTYPE;
V_ALBUME VECTOR_ALBUM := VECTOR_ALBUM();--aici pun albumele - vector
TYPE DETALII_MUZICIAN IS RECORD (
    NUME_MUZICIAN.NUME%TYPE,
    VESTA NUMBER(3),
    POZITIE MEMBRU_TRUPA.POZITIE%TYPE
);
TYPE TABLOU_MEMBRII IS TABLE OF DETALII_MUZICIAN;
T_MEMBRII TABLOU_MEMBRII := TABLOU_MEMBRII(); --TABLOU IMBRICAT CU MUZICIENII

TYPE TABLOU_INSTRUMENTE IS TABLE OF INSTRUMENT.NUME%TYPE INDEX BY INSTRUMENTUTILIZATOR%TYPE;
T_INSTRUMENTE TABLOU_INSTRUMENTE; --TABLOU CU INSTRUMENTE INDEXATE PER POZITIE

VCOUNT NUMBER(3) := 1;
TRUPA TRUPA.ID_TRUPA%TYPE;
INDX INSTRUMENTUTILIZATOR%TYPE;

BEGIN
    --pun albumele unei case de discuri specificate intr-un vector, in ordinea datii lansarii
    FOR I IN (SELECT * FROM ALBUM WHERE ID_CASA_DISCURI = V_CD ORDER BY DATA_LANSARE DESC) LOOP
        V_ALBUME.EXTEND;
        V_ALBUME(VCOUNT) := I;
        VCOUNT := VCOUNT + 1;
    END LOOP;
    VCOUNT := VCOUNT - 1;

    DBMS_OUTPUT.PUT_LINE('*Casa de discuri are ' || V_ALBUME.COUNT || ' albume');

    --pentru cel mai vechi album, gasesc id-ul trupei
    SELECT ID_TRUPA
    INTO TRUPA
    FROM ALBUM
    WHERE ID_CASA_DISCURI = V_CD
    ORDER BY DATA_LANSARE ASC
    LIMIT 1;
    DBMS_OUTPUT.PUT_LINE('Albumul cu numarul ' || V_ALBUME(1).ID_ALBUM || ' este lansat de trupa ' || TRUPA.NUME);

    --pun instrumentele unei trupe specificate intr-un vector, in ordinea datii lansarii
    FOR I IN (SELECT * FROM INSTRUMENT WHERE ID_TRUPA = TRUPA.ID_TRUPA ORDER BY UTILIZATOR%TYPE DESC) LOOP
        T_INSTRUMENTE.EXTEND;
        T_INSTRUMENTE(VCOUNT) := I;
        VCOUNT := VCOUNT + 1;
    END LOOP;
    VCOUNT := VCOUNT - 1;

    DBMS_OUTPUT.PUT_LINE('Trupa ' || TRUPA.NUME || ' foloseste urmatoarele instrumente:');

    FOR I IN 1..VCOUNT LOOP
        DBMS_OUTPUT.PUT_LINE(T_INSTRUMENTE(I).NUME);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('');
    DBMS_OUTPUT.PUT_LINE(' ');
END;

```

Script Output X Task completed in 0.064 seconds

Procedure EX6 compiled

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
--pentru cel mai vechi album, gasesc id-ul trupel
SELECT ID_TRUPA
FROM ALBUM
WHERE ID_ALBUM = V_ALBUM(VCOUNT).ID_ALBUM;
--cum e ordonat, ca sa mai veche e mereu ultima (ca sa folosesc last mai jos si sa aliba sens)
--CEL mai vechi album,
DBMS_OUTPUT.PUT_LINE('Cel mai vechi album: ' || V_ALBUM(V_ALBUM.LAST).NUME );
DBMS_OUTPUT.NEW_LINE;
--tabloiu membrilor
SELECT M_ID, COUNT(M_ID) MONTHS_BETWEEN(SYSDATE,M.ZI_NASTERE)/12 AS VARSTA,
      MT_POZITIE
BULK COLLECT INTO T_MEMBERII
FROM TRUPA T
JOIN MEMBRU_TRUPA MT ON T.ID_TRUPA = MT.ID_TRUPA
JOIN MUZICIAN M ON M.ID_MUZICIAN = MT.ID_MUZICIAN
WHERE T.ID_TRUPA = VTRUPA;
--afisarea membrilor trupei
FOR I IN T_MEMBERII.FIRST .. T_MEMBERII.LAST LOOP
  DBMS_OUTPUT.PUT_LINE(T_MEMBERII(I).NAME);
END LOOP;
--tabel indexat cu UTILIZATORUL cu instrumentele TRUPEI
FOR I IN (SELECT NUME_UTILIZATOR FROM INSTRUMENT WHERE ID_TRUPA = VTRUPA) LOOP
  T_INSTRUMENTE(I.UTILIZATOR) := I.NUME;
END LOOP;
--se suprascriu cele care repetă utilizatorul - ramane 1
DBMS_OUTPUT.NEW_LINE;
--cate un instrument din fiecare pozitie care spune in trupa
--inafara de vocalist (nu are instrumente)
FOR I IN (SELECT UNIQUE POZITIE FROM MEMBRU_TRUPA WHERE ID_TRUPA = VTRUPA AND POZITIE != 'Vocalist') LOOP
  DBMS_OUTPUT.PUT_LINE(I.POZITIE || ' ' || T_INSTRUMENTE(I.POZITIE));
  T_INSTRUMENTE.DELETE(I.POZITIE);
END LOOP;
PL/SQL procedure successfully completed.
```

```
--pentru instrumentele cu pozitii de la 1 la 100
FOR I IN (SELECT NUME_UTILIZATOR FROM INSTRUMENT WHERE ID_TRUPA = VTRUPA) LOOP
  T_INSTRUMENTE(I.UTILIZATOR) := I.NUME;
END LOOP;
--se suprascriu cele care repetă utilizatorul - ramane 1
DBMS_OUTPUT.NEW_LINE;
--cate un instrument din fiecare pozitie care spune in trupa
--inafara de vocalist (nu are instrumente)
FOR I IN (SELECT UNIQUE POZITIE FROM MEMBRU_TRUPA WHERE ID_TRUPA = VTRUPA AND POZITIE != 'Vocalist') LOOP
  DBMS_OUTPUT.PUT_LINE(I.POZITIE || ' ' || T_INSTRUMENTE(I.POZITIE));
  T_INSTRUMENTE.DELETE(I.POZITIE);
END LOOP;
--daca mai sunt instrumente
IF T_INSTRUMENTE.COUNT > 0 THEN
  DBMS_OUTPUT.NEW_LINE();
  DBMS_OUTPUT.PUT_LINE('Există instrumente pentru pozitii neocupate: ');
  --itera prin ele pana nu mai am index
  INDEX := T_INSTRUMENTE.FIRST;
  WHILE INDEX IS NOT NULL LOOP
    DBMS_OUTPUT.PUT_LINE(INDEX || ' ' || T_INSTRUMENTE(INDEX));
    INDEX := T_INSTRUMENTE.NEXT(INDEX);
  END LOOP;
END IF;
EXCEPTION
  WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20000,'nu au fost gasite date');
  WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost returnate prea multe linii');
  WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002,'alta eroare');
END;
/
-----
/
BEGIN
  EX6(5);
END;
```

---

**7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.**

---

➤ **Cerință:**

Pentru fiecare trupă din baza de date (luate în ordine alfabetică), să se afișeze numele ei, numărul de membri și top 3 cântece (sau câte există, dacă sunt mai puțin de 3) (top-ul se ia în ordine descrescătoare de lungime).

➤ Cod sursă:

```
--7
--pentru fiecare trupa, luate in ordine alfabetica, sa se afiseze numele si
--numarul de membri
--sa se afiseze top 3 cantece (sau cate exista, daca are mai putin de 3) (in
--ordine de la cele mai lungi cantece)
CREATE OR REPLACE PROCEDURE EX7 AS

CURSOR TRUPE IS --cursor explicit
    SELECT ID_TRUPA, NUME,
        (SELECT COUNT(*) FROM MEMBRU_TRUPA MT WHERE MT.ID_TRUPA =
T.ID_TRUPA GROUP BY T.ID_TRUPA) AS NR_MEMBRII
        FROM TRUPA T;

CURSOR TOP_CANTECE(V_ID TRUPA.ID_TRUPA%TYPE) IS --cursor explicit,
parametrizat
    SELECT * FROM(
        SELECT NUME
        FROM CANTEC
        JOIN TRUPA_CANTEC ON CANTEC.ID_CANTEC = TRUPA_CANTEC.ID_CANTEC
        WHERE V_ID = ID_TRUPA
        ORDER BY LUNGIME DESC)
    WHERE ROWNUM <= 3;

NUME_CANTEC CANTEC.NUME%TYPE; --ca sa fac fetch

BEGIN

FOR VTRUPA IN TRUPE LOOP --parcurgere cu for
    DBMS_OUTPUT.PUT_LINE(VTRUPA.NUME);
    IF VTRUPA.NR_MEMBRII > 1 THEN
        DBMS_OUTPUT.PUT_LINE('Trupa are ' || VTRUPA.NR_MEMBRII || '
membri');
    ELSIF VTRUPA.NR_MEMBRII = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Trupa are un membru');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Trupa nu are muzicieni inregistrati');

    END IF;
END LOOP;
END;
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
END IF;
--parcurgere cu open, close si fetch
OPEN TOP_CANTECE(VTRUPA.ID_TRUPA);
LOOP
    FETCH TOP_CANTECE INTO NUME_CANTEC;
    EXIT WHEN TOP_CANTECE%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE(TOP_CANTECE%ROWCOUNT || ' . ' || 
NUME_CANTEC);
    END LOOP;
    CLOSE TOP_CANTECE;
    DBMS_OUTPUT.NEW_LINE;
END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20000,'nu au fost
gasite date');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost
returnate prea multe linii');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002,'alta eroare');

END;
/
/
BEGIN
    EX7;
END;
```

➤ Printscrean:

```
--sa se afiseze top 3 cantece (sau cate exista, daca are mai putin de 3) (in ordine de la cele mai lungi cantece)
CREATE OR REPLACE PROCEDURE EX7 AS

CURSOR TRUPE IS --cursor explicit
SELECT ID_TRUPA, NUME,
       (SELECT COUNT(*) FROM MEMBRU_TRUPA MT WHERE MT.ID_TRUPA = T.ID_TRUPA GROUP BY T.ID_TRUPA) AS NR_MEMBRII
FROM TRUPA T;

CURSOR TOP_CANTECE(V_ID TRUPA.ID_TRUPA%TYPE) IS --cursor explicit, parametrizat
SELECT * FROM(
    SELECT NUME
    FROM CANTEC
    JOIN TRUPA_CANTEC ON CANTEC.ID_CANTEC = TRUPA_CANTEC.ID_CANTEC
    WHERE V_ID = ID_TRUPA
    ORDER BY LUNGIME DESC)
WHERE ROWNUM <= 3;

NUME_CANTEC CANTEC.NUME%TYPE; --ca sa fac fetch

BEGIN

FOR VTRUPA IN TRUPE LOOP --parcureg cu for
    DBMS_OUTPUT.PUT_LINE(VTRUPA.NUME);
    IF VTRUPA.NR_MEMBRII > 1 THEN
        DBMS_OUTPUT.PUT_LINE('Trupa are ' || VTRUPA.NR_MEMBRII || ' membrii');
    ELSIF VTRUPA.NR_MEMBRII = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Trupa nu are un membru');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Trupa nu are muzicieni inregistrati');
    END IF;
--parcureg cu open, close si fetch
OPEN TOP_CANTECE(VTRUPA.ID_TRUPA);
LOOP
    FETCH TOP_CANTECE INTO NUME_CANTEC;
    EXIT WHEN TOP_CANTECE%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE(TOP_CANTECE%ROWCOUNT || ' . ' || NUME_CANTEC);
END LOOP;
CLOSE TOP_CANTECE;
DBMS_OUTPUT.NEW_LINE;
END LOOP;

END;
```

Script Output | Query Result x  
Task completed in 0.057 seconds

Procedure EX7 compiled

```
Qr/ red Aa == recordlabel x
Qr/ CT Replace Skip Replace All recordlabel x
Worksheet Query Builder recordlabel x
recordlabel x Buffer Size 20000
recordlabel x
My Chemical Romance
Trupa are 4 membrii
1. I Never Told You What I Do for a Living
2. Thank You for the Venom
3. Helena

Gerard Way
Trupa are un membru
1. No Shows

Frank Iero
Trupa are un membru
1. Violence
2. Sewerwolf

The Future Violents
Trupa are 2 membrii
1. Record Ender
2. Violence
3. Sewerwolf

TWRP
Trupa are 4 membrii
1. A Human's Touch
2. HTMLove
3. The No Pants Dance

Tom Cardy
Trupa are un membru
1. Online

Queen
Trupa are 4 membrii
1. Bohemian Rhapsody
2. Spread Your Wings
3. I Want To Break Free

Will Wood
Trupa nu are muzicieni inregistrati

The Tapeworms
Trupa nu are muzicieni inregistrati

Aesnrikko
Trupa nu are muzicieni inregistrati

Montaigne
Trupa are un membru
1. Online

NSP
Trupa nu are muzicieni inregistrati
1. The No Pants Dance
```

PL/SQL procedure successfully completed.

Compiler - Log

---

**8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție.**

---

care să utilizeze **într-o singură comandă SQL** 3 dintre tabelele create. Tratați toate excepțiile care pot apărea, incluzând excepțiile predefinite NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați **toate** cazurile tratate

➤ **Cerință:**

Pentru un id de casa de discuri dat, scrieți o funcție în cadrul careia să afișați ziua săptămânii în care au fost semnate cele mai multe contracte și afișați mesaj corespunzător dacă toate contractele sunt expirate. Funcția returnează numărul de muzicieni care au ziua de nastere în aceeași zi a săptămânii cu cea gasită anterior. Ridicați eroare dacă casa de discuri nu are contracte semnate, dar există, sau dacă nu există casa de discuri.

➤ Cod sursă:

```
--8
/
CREATE OR REPLACE FUNCTION EX8 (V_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE
DEFAULT 1) --se da id-ul casei de discuri
RETURN NUMBER IS

    NR_MUZICIENI NUMBER(3);

    TYPE ZILE_SAPTMANA IS TABLE OF VARCHAR2(10) INDEX BY PLS_INTEGER;
    ZILE_SAPT ZILE_SAPTMANA := ZILE_SAPTMANA();

    ZI VARCHAR2(10); --aici o să punem ziua gasita
    VCOUNT NUMBER(3) := 0;
    NR_EXPIRAT NUMBER(3) := 0;
    NR_CONTRACTE NUMBER(3) := 0;
    VMAXIM NUMBER(3) := 0;

    CURSOR CONTRACTE (ZI_SAPT PLS_INTEGER, V_ID
CASA_DISCURI.ID_CASA_DISCURI%TYPE) IS
        SELECT --cursor pt contractele semnante in acea zi
        CASE WHEN MONTHS_BETWEEN(DATA_EXPIRARE, SYSDATE) < 0 THEN 1
        ELSE 0 END AS EXPIRAT --1 INSEAMNA EXPIRAT
        FROM CONTRACT
        WHERE ID_CASA_DISCURI = V_ID
        AND TO_CHAR(DATA_SEMNARE, 'D') = ZI_SAPT;

    NO_CONTRACT EXCEPTION;
    NO_CD EXCEPTION;

BEGIN
    --zilele săptămânii
    ZILE_SAPT (1) := 'LUNI'; ZILE_SAPT (2) := 'MARTI'; ZILE_SAPT (3) :=
    'MIERCURI'; ZILE_SAPT (4) := 'JOI';
    ZILE_SAPT (5) := 'VINERI'; ZILE_SAPT (6) := 'SAMBATA'; ZILE_SAPT
    (7) := 'DUMINICA';

    --verific dacă există casa de discuri data
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
SELECT COUNT(*)
INTO VCOUNT
FROM CASA_DISCURI
WHERE ID_CASA_DISCURI = V_ID;

IF VCOUNT = 0 THEN
    RAISE NO_CD;
END IF;

--verific daca are contracte
SELECT COUNT(*)
INTO VCOUNT
FROM CONTRACT
WHERE ID_CASA_DISCURI = V_ID;

IF VCOUNT = 0 THEN
    RAISE NO_CONTRACT;
END IF;

--CAUT ziua cu nr maxim de contracte semnate
FOR I IN 1..7 LOOP --iterez prin fiecare zi
    VCOUNT := 0;
    FOR CONTR IN CONTRACTE(I, V_ID) LOOP
        VCOUNT := VCOUNT + 1;
        NR_CONTRACTE := NR_CONTRACTE + 1; --AICI LE NUMAR PE TOATE
        NR_EXPIRAT := NR_EXPIRAT + CONTR.EXPIRAT; --AICI NR CELE
    EXPIRATE
    END LOOP; --ptr fiecare contract semnat in ziua aia, adun 1
    IF VCOUNT > VMAXIM THEN
        VMAXIM := VCOUNT;
        ZI := I;
    END IF;
END LOOP;
--OK! acum am numarul zilei in variabila ZI
DBMS_OUTPUT.PUT_LINE('Cele mai multe contracte au fost semnate intr-o
zi de ' || ZILE_SAPTA(ZI));
--afisati mesaj corespunzator daca toate contractele sunt expirate
IF NR_EXPIRAT = NR_CONTRACTE THEN
    DBMS_OUTPUT.PUT_LINE('Toate contractele casei de discuri sunt
expirate!');
END IF;

-----
--returneaza numarul de muzicieni care a ziua de nastere in aceasi zi
a saptamanii
--aici o sa folosesc 3 tabele intr-o comanda sql
SELECT COUNT(*)
INTO NR_MUZICIENI
FROM CASA_DISCURI CD
JOIN CONTRACT C ON C.ID_CASA_DISCURI = CD.ID_CASA_DISCURI
JOIN MUZICIAN M ON M.ID_MUZICIAN = C.ID_MUZICIAN
WHERE CD.ID_CASA_DISCURI = V_ID
    AND TO_CHAR(M.ZI_NASTERE, 'D') = ZI;

RETURN NR_MUZICIENI;

EXCEPTION
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
        WHEN NO_CD THEN RAISE_APPLICATION_ERROR (-20003,'nu au fost gasita
casa de discuri cu acel id');
        WHEN NO_CONTRACT THEN RAISE_APPLICATION_ERROR (-20004,'nu au fost
gasite contracte la acea casa de discuri');
        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20000,'nu au fost
gasite date');
        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost
returnate prea multe linii');
        WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002,'alta eroare');

END EX8;
/
-----
/
DECLARE
    NR_MUZICIENI NUMBER(3);
BEGIN
    NR_MUZICIENI := EX8(4);
    IF NR_MUZICIENI = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista muzicieni de la aceasta casa de
discuri care indeplineasca conditiile');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Există '||NR_MUZICIENI|| ' muzicieni cu ziua
de nastere in aceasi zi');
    END IF;
END;
/
```

➤ Printscrean

```

1 -->_0_
2 /
3 CREATE OR REPLACE FUNCTION EXB (V_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE DEFAULT 1) --se da id-ul casei de discuri
4 RETURN NUMBER IS
5
6  NR_MUZICIENI NUMBER(5);
7
8  TYPE ZILE_SAPTMANA IS TABLE OF VARCHAR2(10) INDEX BY PLS_INTEGER;
9  ZILE_SAPT ZILE_SAPTMANA := ZILE_SAPTMANA();
10
11  ZI VARCHAR2(10); --aici o sa punem ziua gasita
12  VCOUNT NUMBER(3) := 0;
13  NR_EXPIRAT NUMBER(3) := 0;
14  NR_CONTRACTE NUMBER(3) := 0;
15  VMAXIM NUMBER(3) := 0;
16
17  CURSOR CONTRACTE (ZI_SAPT PLS_INTEGER, V_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE) IS
18    SELECT --cursor pt contractele semnate in acea zi
19      CASE WHEN MONTHS_BETWEEN(DATA_EXPIRARE,SYSDATE) < 0 THEN 1 ELSE 0 END AS EXPIRAT --1 inseamna expirat
20    FROM CONTRACT
21    WHERE ID_CASA_DISCURI = V_ID
22    AND TO_CHAR(DATA_SEMNARE,'D') = ZI_SAPT;
23
24  NO_CONTRACT EXCEPTION;
25  NO_CD EXCEPTION;
26
27  BEGIN
28    --zilele saptamanii
29    ZILE_SAPT (1) := 'LUNI'; ZILE_SAPT (2) := 'MARTI'; ZILE_SAPT (3) := 'MIERCIURI'; ZILE_SAPT (4) := 'JOI';
30    ZILE_SAPT (5) := 'VINERI'; ZILE_SAPT (6) := 'SAMBATA'; ZILE_SAPT (7) := 'DUMINICA';
31
32    --verific daca exista casa de discuri data
33    SELECT COUNT(*) INTO VCOUNT
34    FROM CASA_DISCURI
35    WHERE ID_CASA_DISCURI = V_ID;
36
37  EXCEPTION
38    WHEN NO_CD THEN
39      RAISE NO_CD;
40    WHEN NO_CONTRACT THEN
41      RAISE NO_CONTRACT;
42  END;
43
44  PL/SQL procedure successfully completed.

```

```

1 -->_0_
2 /
3 CREATE OR REPLACE PROCEDURE EXB (V_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE)
4 AS
5
6  IF VCOUNT = 0 THEN
7    RAISE NO_CD;
8  END IF;
9
10
11  --verific daca are contracte
12  SELECT COUNT(*)
13  INTO VCOUNT
14  FROM CONTRACT
15  WHERE ID_CASA_DISCURI = V_ID;
16
17  IF VCOUNT = 0 THEN
18    RAISE NO_CONTRACT;
19  END IF;
20
21
22  --CAUT ziua cu nr maxim de contracte semnate
23  FOR I IN 1..7 LOOP --iterez prin fiecare zi
24    VCOUNT := 0;
25    FOR CONTR IN CONTRACTE(I, V_ID) LOOP
26      VCOUNT := VCOUNT + 1;
27      NR_CONTRACTE := NR_CONTRACTE + 1; --AICI LE NUMAR PE TOATE
28      NR_EXPIRAT := NR_EXPIRAT + CONTR.EXPIRAT; --AICI NR CELE EXPIRANTE
29    END LOOP; --pt fiecare contract semnat in ziua aia, adun 1
30    IF VCOUNT > VMAXIM THEN
31      VMAXIM := VCOUNT;
32      ZI := I;
33    END IF;
34  END LOOP;
35
36  --OK! acum am numarul zilei in variabila ZI
37  DBMS_OUTPUT.PUT_LINE('Cele mai multe contracte au fost semnante intr-o zi de ' || ZILE_SAPT(ZI));
38
39  --afisati mesaj corespunzator daca toate contractele sunt expirate
40  IF NR_EXPIRAT = NR_CONTRACTE THEN
41    DBMS_OUTPUT.PUT_LINE('Toate contractele casei de discuri sunt expirate!');
42  END IF;
43
44
45  --returneaza numarul de muzicieni care a ziua de nastere in aceasi zi a saptamanii
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

```

Serba Ioana-Raluca  
Seria 24 Grupa 241

The screenshot shows the Oracle SQL Developer interface with a PL/SQL procedure named "recordlabel". The code implements a function to calculate the number of musicians born on a specific day of the week. It uses three tables: NR\_MUZICIENI, CASA\_DISCURI, and MUZICIAN. The function handles various error cases like no contract or too many rows. The output window displays messages about the most contracts per day of the week and the existence of musicians born on that day.

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```

The screenshot shows a PL/SQL block. It declares a variable "NR\_MUZICIENI" of type NUMBER(3). The block begins with an IF statement checking if the value is 0. If true, it prints a message about the absence of musicians. If false, it prints a message about the existence of musicians born on the same day. The block ends with an END IF; statement.

```
ERROR at line 1:
ORA-20003: nu a fost gasita casa de discuri cu acel id
ORA-06512: at "LUCA.EX8", line 84
```

The screenshot shows the Oracle SQL Developer interface. In the top pane, a PL/SQL script is displayed:

```
96  /
97  DECLARE
98      NR_MUZICIENI NUMBER(3);
99  BEGIN
100     NR_MUZICIENI := EX8(21);
101    IF NR_MUZICIENI = 0 THEN
102        DBMS_OUTPUT.PUT_LINE('Nu exista muzicieni de la aceasta casa de discuri care indepl');
103    ELSE
104        DBMS_OUTPUT.PUT_LINE('Există '||NR_MUZICIENI|| ' muzicieni cu ziua de nastere i');
105    END IF;
106  END;
107  /
108 --ca sa exemplific si exceptia de NO_CONTRACT, o sa inserez o noua casa de discuri fara
109 INSERT INTO CASA_DISCURSI (NUME, WEBSITE, AN_INFIINTARE)
110 VALUES ('Angel Records', 'fallenangel.com', 2014);
111
```

In the bottom pane, the execution results are shown:

```
DECLARE
*
ERROR at line 1:
ORA-20004: nu au fost gasite contracte la aceea casa de discuri
ORA-06512: at "LUCA.EX8", line 85
ORA-06512: at line 4

https://docs.oracle.com/error-help/db/ora-20004/
```

The screenshot shows the Oracle SQL Developer interface with the DBMS Output window open. The window title is "Dbms Output" and it has a buffer size of 20000. The output content is:

```
recordlabel x
Cele mai multe contracte au fost semnate intr-o zi de JOI
Există 1 muzicieni cu ziua de nastere în aceasi zi

Cele mai multe contracte au fost semnate intr-o zi de DUMINICA
Toate contractele casei de discuri sunt expirate!
Nu există muzicieni de la aceasta casa de discuri care indeplineasca conditiile

Cele mai multe contracte au fost semnate intr-o zi de MARTI
Toate contractele casei de discuri sunt expirate!
Există 1 muzicieni cu ziua de nastere în aceasi zi
```

---

**9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură.**

---

care să aibă minim 2 parametri și să utilizeze **într-o singură comandă SQL** 5 dintre tabelele create. Definiți minim 2 exceptii proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați **toate** cazurile definite și tratate.

➤ **Cerință:**

Definiți o procedura care afiseaza numele celui mai lung (ca și durata) album al unei case de discuri, al cărei id se da ca primul parametru. Se mai afiseaza și orice alta trupa care a contribuit la orice cantec care se află pe album, dacă există. Se întoarce prin parametrii numele trupei care a lansat albumul mentionat și numele celui mai scurt cantec ca și durată. Se ridică eroare dacă casa de discuri nu există, dacă există și nu are albume, dacă există 2 sau mai multe albume de aceeași lungime maximă la acea casa de discuri sau dacă există 2 sau mai multe cantece de lungime minima pe acel album.

➤ Cod sursă:

➤ --9

```
--prințr-un parametru se introduce id-ul unei case de discuri
--eroare dacă nu există casa de discuri, sau dacă casa nu are albume

--al doilea se întoarce numele trupei care a lansat cel mai lung album
--după lungimea canticelor continue
--și al treilea parametru = cel mai scurt cantec de pe acel album
--se ridică eroare dacă există 2 albume de exact aceeași lungime maximă
--sau 2+ cantece de cea mai scurtă lungime

--se afisează în procedura numele albumului și dacă mai există alți
--muzicieni care au
--contribuit la cantecele de pe el în afară de trupa care l-a lansat

/
CREATE OR REPLACE PROCEDURE EX9
    (V_ID IN CASA_DISCURI.ID_CASA_DISCURI%TYPE, --ID CASA DE DISCURSI
     CARE VA FI DAT
        NUME_TRUPA OUT TRUPA.NUME%TYPE,
        NUME_CANTEC OUT CANTEC.NUME%TYPE) AS

    --cursor pentru toate trupele care au contribuit la un album care
    --să fie diferite de trupa care l-a lansat
    --ca să folosească într-o singură comandă SQL 5 dintre tabelele
    --create
    --voi face cu numele albumului, în loc de id_album
    CURSOR FEAT_TRUPE (NUME_ALBUM ALBUM.NUME%TYPE, NUME_TRUPA
    TRUPA.NUME%TYPE) IS
        SELECT unique T.NUME
        FROM TRUPA T
        JOIN TRUPA_CANTEC TC ON TC.ID_TRUPA = T.ID_TRUPA
```

```
JOIN CANTEC C ON TC.ID_CANTEC = C.ID_CANTEC
JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
JOIN ALBUM A ON CA.ID_ALBUM = A.ID_ALBUM
WHERE A.NUME = NUME_ALBUM
AND T.NUME != NUME_TRUPA;

VCOUNT NUMBER(2);
VALBUM ALBUM.ID_ALBUM%TYPE;
NUME_ALBUM ALBUM.NUME%TYPE;
MLUNGIME CANTEC.LUNGIME%TYPE;

-- exceptii
NO_CASA_DISCURI EXCEPTION; --cand nu exista cd
NO_ALBUMS EXCEPTION; --cand exista cd si nu are albume
TOO_MANY_ALBUMS EXCEPTION; --CAND SUNT MAI MULTE ALBUME DE LUNGIME MAXIMA
TOO_MANY_CANTECE EXCEPTION; --CAND SUNT MAI MULTE CANTECE DE LUNGIME MINIMA

BEGIN
--daca exista cd
SELECT COUNT(*)
INTO VCOUNT
FROM CASA_DISCURI
WHERE V_ID = ID_CASA_DISCURI;

IF VCOUNT < 1 THEN
    RAISE NO_CASA_DISCURI;
END IF;

--daca are albume
SELECT COUNT(*)
INTO VCOUNT
FROM ALBUM
WHERE V_ID = ID_CASA_DISCURI;

IF VCOUNT < 1 THEN
    RAISE NO_ALBUMS;
END IF;

--se ridica eroare daca exista 2+ albume de exact aceasi lungime maxima
--la acea casa de discuri
SELECT MAX(SUM(LUNGIME))
INTO MLUNGIME
FROM CANTEC C
JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
JOIN ALBUM A ON A.ID_ALBUM = CA.ID_ALBUM
WHERE A.ID_CASA_DISCURI = V_ID
GROUP BY CA.ID_ALBUM;

SELECT COUNT(*)
INTO VCOUNT
FROM ALBUM A
WHERE V_ID = ID_CASA_DISCURI
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
        AND (SELECT SUM(LUNGIME)
      FROM CANTEC C
      JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
      WHERE CA.ID_ALBUM = A.ID_ALBUM
      GROUP BY ID_ALBUM) = MLUNGIME;

      IF VCOUNT > 1 THEN
          RAISE TOO_MANY_ALBUMS;
      END IF;

--id cel mai lung album
SELECT ID_album, NUME
INTO VALBUM, NUME_ALBUM
FROM ALBUM A
WHERE V_ID = ID_CASA_DISCURI
AND (SELECT SUM(LUNGIME)
      FROM CANTEC C
      JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
      WHERE CA.ID_ALBUM = A.ID_ALBUM
      GROUP BY ID_ALBUM) = MLUNGIME;

--trupa celui mai lung album
SELECT T.NUME
INTO NUME_TRUPA
FROM TRUPA T
JOIN ALBUM A ON A.ID_TRUPA = T.ID_TRUPA
WHERE ID_ALBUM = VALBUM;

--se ridica eroare daca exista 2+ cantece de cea mai scurta lungime
SELECT MIN(LUNGIME)
INTO MLUNGIME --LUNGIMEA minima de cantic
FROM CANTEC C
JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
WHERE ID_ALBUM = VALBUM;

SELECT COUNT(*)
INTO VCOUNT
FROM CANTEC C
JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
WHERE ID_ALBUM = VALBUM
AND LUNGIME = MLUNGIME;

IF VCOUNT > 1 THEN
    RAISE TOO_MANY_CANTECE;
END IF;

--eu si vcount suntem bffs <3 i love vcount

--cel mai scurt cantic
SELECT NUME
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
INTO NUME_CANTEC
FROM CANTEC C
JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
WHERE ID_ALBUM = VALBUM
AND LUNGIME = MLUNGIME;

--se afiseaza in procedura numele albumului
DBMS_OUTPUT.PUT_LINE('Cel mai lung album: ');
DBMS_OUTPUT.PUT_LINE(NUME_ALBUM);
DBMS_OUTPUT.PUT_LINE('-----');

--daca mai exista alti muzicieni care au
--contribuit la cantecele de pe el inafara de trupa care l-a lansat
= afisam
FOR VTRUPA IN FEAT_TRUPE(NUME_ALBUM,NUME_TRUPA) LOOP
    DBMS_OUTPUT.PUT_LINE('feat: ' || VTRUPA.NUME);
    DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;

EXCEPTION
    WHEN NO_CASA_DISCURI THEN RAISE_APPLICATION_ERROR (-20003,'nu a
fost gasita casa de discuri cu acel id');
    WHEN NO_ALBUMS THEN RAISE_APPLICATION_ERROR (-20004,'nu au fost
gasite albume la acea casa de discuri');
    WHEN TOO_MANY_ALBUMS THEN RAISE_APPLICATION_ERROR (-20005,'au fost
gasite prea multe albume cu lungime maxima');
    WHEN TOO_MANY_CANTECE THEN RAISE_APPLICATION_ERROR (-20006,'au fost
gasite prea multe cantece cu lungime minima');

    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20000,'nu au fost
gasite date');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost
returnate prea multe linii');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002,'alta eroare');

END;
/

-----
/
DECLARE
    NUME_TRUPA TRUPA.NUME%TYPE;
    NUME_CANTEC CANTEC.NUME%TYPE;
BEGIN
    EX9(1,NUME_TRUPA,NUME_CANTEC);
    DBMS_OUTPUT.PUT_LINE('Trupa care a lansat albumul: '|| NUME_TRUPA);
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('Cel mai scurt cantec de pe album: '|| 
NUME_CANTEC);
    DBMS_OUTPUT.new_line;
END;
```

➤ Printscrean

```
CREATE OR REPLACE PROCEDURE EX9
( _ID IN CASA_DISCURSI.ID_CASA_DISCURSI%TYPE, --ID CASA DE DISCURSI CARE VA FI DAT
  NUME_TRUPA OUT TRUPA.NUME%TYPE,
  NUME_CANTEC OUT CANTEC.NUME%TYPE ) AS
BEGIN
  --cursor pentru toate trupele care au contribuit la un album care sa fie diferite de trupa care l-a lansat
  --ca sa folosesc intr-o singură comandă SQL 5 dintre tabelele create
  --voi face cu numele albumului, in loc de id_album
  CURSOR FEAT_TRUPE (NUME_ALBUM ALBUM.NUME%TYPE, NUME_TRUPA TRUPA.NUME%TYPE) IS
    SELECT unique T.NUME
    FROM TRUPA T
    JOIN CANTEC_C ON TC.ID_CANTEC = T.ID_TRUPA
    JOIN CANTEC C ON TC.ID_CANTEC = C.ID_CANTEC
    JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
    JOIN ALBUM A ON CA.ID_ALBUM = A.ID_ALBUM
    WHERE A.NUME = NUME_ALBUM
    AND T.NUME != NUME_TRUPA;
  VCOUNT NUMBER(2);
  FOR FEAT IN FEAT_TRUPE
  LOOP
    VCOUNT := VCOUNT + 1;
  END LOOP;
  IF VCOUNT > 1 THEN
    RAISE NO_CASA_DISCURSI;
  END IF;
END;

--data are albume
SELECT COUNT(*)
INTO VCOUNT
FROM ALBUM
WHERE V_ID = ID_CASA_DISCURSI;
IF VCOUNT < 1 THEN
  RAISE NO_ALBUMS;
END IF;
--use sfidura encara data există ?a albume de având această lungime maximă
```

```
Cel mai lung album:
Three Cheers For Sweet Revenge
-----
Trupa care a lansat albumul: My Chemical Romance
-----
Cel mai scurt cantică de pe album: Interlude

Cel mai lung album:
Digital Nightmare
-----
feat: Tom Cardy
-----
Trupa care a lansat albumul: TWRP
-----
Cel mai scurt cantică de pe album: Digital Nightmare
```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```

55 --la acea casa de discuri
56 SELECT MAX(SUM(LUNGIME))
57 INTO MLUNGIME
58 FROM CANTEC C
59 JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
60 JOIN ALBUM A ON A.ID_ALBUM = CA.ID_ALBUM
61 WHERE A.ID_CASA_DISCURI = V_ID
62 GROUP BY CA.ID_ALBUM;
63
64
65 SELECT COUNT(*)
66 INTO VCOUNT
67 FROM ALBUM A
68 WHERE V_ID = ID_CASA_DISCURI
69 AND (SELECT SUM(LUNGIME)
70 FROM CANTEC C
71 JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
72 WHERE CA.ID_ALBUM = A.ID_ALBUM
73 GROUP BY ID_ALBUM) = MLUNGIME;
74
75
76 IF VCOUNT > 1 THEN
77     RAISE TOO_MANY_ALBUMS;
78 END IF;
79
80
81 --id cel mai lung album
82 SELECT ID_album, NUME
83 INTO VALBUM, NUME_ALBUM
84 FROM ALBUM A
85 WHERE V_ID = ID_CASA_DISCURI
86 AND (SELECT SUM(LUNGIME)
87 FROM CANTEC C
88 JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
89 WHERE CA.ID_ALBUM = A.ID_ALBUM
90 GROUP BY ID_ALBUM) = MLUNGIME;
91
92
93 --trupa celui mai lung album
94 SELECT T.NUME
95 INTO NUME_TRUPA
96 FROM TRUPA T
97 JOIN ALBUM A ON A.ID_TRUPA = T.ID_TRUPA
98 WHERE ID_ALBUM = VALBUM;
99
100
101 --se ridica eroare daca exista 2+ cantece de cea mai scurta lungime
102 SELECT MIN(LUNGIME)
103 INTO MLUNGIME --LUNGIMEA minima de cantece
104 FROM CANTEC C
105 JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
106 WHERE ID_ALBUM = VALBUM;
107
108
109
110 SELECT COUNT(*)
111 INTO VCOUNT
112 FROM CANTEC C

```

Script Output X | Task completed in 0.071 seconds

\*

ERROR at line 1:  
ORA-20003: nu a fost gasita casa de discuri cu acel id  
ORA-06512: at "LUCA.EX9", line 146  
ORA-06512: at line 5

<https://docs.oracle.com/error-help/db/ora-20003/>

```

77 RAISE TOO_MANY_ALBUMS;
78 END IF;
79
80
81 --id cel mai lung album
82 SELECT ID_album, NUME
83 INTO VALBUM, NUME_ALBUM
84 FROM ALBUM A
85 WHERE V_ID = ID_CASA_DISCURI
86 AND (SELECT SUM(LUNGIME)
87 FROM CANTEC C
88 JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
89 WHERE CA.ID_ALBUM = A.ID_ALBUM
90 GROUP BY ID_ALBUM) = MLUNGIME;
91
92
93 --trupa celui mai lung album
94 SELECT T.NUME
95 INTO NUME_TRUPA
96 FROM TRUPA T
97 JOIN ALBUM A ON A.ID_TRUPA = T.ID_TRUPA
98 WHERE ID_ALBUM = VALBUM;
99
100
101 --se ridica eroare daca exista 2+ cantece de cea mai scurta lungime
102 SELECT MIN(LUNGIME)
103 INTO MLUNGIME --LUNGIMEA minima de cantece
104 FROM CANTEC C
105 JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
106 WHERE ID_ALBUM = VALBUM;
107
108
109
110 SELECT COUNT(*)
111 INTO VCOUNT
112 FROM CANTEC C

```

Script Output X | Task completed in 0.067 seconds

\*

ERROR at line 1:  
ORA-20004: nu au fost gasite albume la acea casa de discuri  
ORA-06512: at "LUCA.EX9", line 147  
ORA-06512: at line 5

<https://docs.oracle.com/error-help/db/ora-20004/>

Serba Ioana-Raluca  
Seria 24 Grupa 241

```

109
110  SELECT COUNT(*)
111    INTO VCOUNT
112   FROM CANTEC C
113  JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
114 WHERE ID_ALBUM = VALBUM
115 AND LUNGIME = MLUNGIME;
116
117 IF VCOUNT > 1 THEN
118   RAISE TOO_MANY_CANTECE;
119 END IF;
120
121
122 --eu si vcount suntem bffs <3 i love vcount
123
124 --cel mai scurt cantec
125  SELECT NUME
126    INTO NUME_CANTEC
127   FROM CANTEC C
128  JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
129 WHERE ID_ALBUM = VALBUM
130 AND LUNGIME = MLUNGIME;
131
132
133 --se afiseaza in procedura numele albumului
134 DBMS_OUTPUT.PUT_LINE('Cel mai lung album: ');
135 DBMS_OUTPUT.PUT_LINE(NUME_ALBUM);
136 DBMS_OUTPUT.PUT_LINE('-----');
137
138 --daca mai exista alti muzicieni care au
139 --contribuit la cantecele de pe el inafara de trupa care l-a lansat = afisam
140 FOR VTRUPA IN FEAT_TRUPE(NUME_ALBUM,NUME_TRUPA) LOOP
141   DBMS_OUTPUT.PUT_LINE('feat: ' || VTRUPA.NUME);
142   DBMS_OUTPUT.PUT_LINE('-----');
143 END LOOP;
144

```

Script Output x | Query Result x

| Task completed in 0.053 seconds

\*

ERROR at line 1:

ORA-20005: au fost gasite prea multe albume cu lungime maxima

ORA-06512: at "LUCA.EX9", line 148

ORA-06512: at line 5

<https://docs.oracle.com/error-help/db/ora-20005/>

```

--se afiseaza in procedura numele albumului
DBMS_OUTPUT.PUT_LINE('Cel mai lung album: ');
DBMS_OUTPUT.PUT_LINE(NUME_ALBUM);
DBMS_OUTPUT.PUT_LINE('-----');

--daca mai exista alti muzicieni care au
--contribuit la cantecele de pe el inafara de trupa care l-a lansat = afisam
FOR VTRUPA IN FEAT_TRUPE(NUME_ALBUM,NUME_TRUPA) LOOP
  DBMS_OUTPUT.PUT_LINE('feat: ' || VTRUPA.NUME);
  DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;

EXCEPTION
  WHEN NO_CASA_DISCURI THEN RAISE_APPLICATION_ERROR (-20003,'nu a fost gasita casa de discuri cu acel id');
  WHEN NO_ALBUMS THEN RAISE_APPLICATION_ERROR (-20004,'nu au fost gasite albume la acea casa de discuri');
  WHEN TOO_MANY_ALBUMS THEN RAISE_APPLICATION_ERROR (-20005,'au fost gasite prea multe albume cu lungime maxima');
  WHEN TOO_MANY_CANTECE THEN RAISE_APPLICATION_ERROR (-20006,'au fost gasite prea multe cantec ce cu lungime minima');

  WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20000,'nu au fost gasite date');
  WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost returnate prea multe linii');
  WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002,'alta eroare');

END;
/
-----/
DECLARE
  NUME_TRUPA TRUPA.NUME%TYPE;
  NUME_CANTEC CANTEC.NUME%TYPE;
BEGIN
  EX9(8,NUME_TRUPA,NUME_CANTEC);
  DBMS_OUTPUT.PUT_LINE('Trupa care a lansat albumul: '|| NUME_TRUPA);
  DBMS_OUTPUT.PUT_LINE('-----');
  DBMS_OUTPUT.PUT_LINE('Cel mai scurt cantec de pe album: '|| NUME_CANTEC);
  DBMS_OUTPUT.new_line;
END;

```

Script Output x | Query Result x

| Task completed in 0.04 seconds

\*

ERROR at line 1:

ORA-20006: au fost gasite prea multe cantec ce cu lungime minima

ORA-06512: at "LUCA.EX9", line 149

---

## 10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

---

➤ **Cerință:**

Definiti un trigger de tip LMD la nivel de comanda care, pentru tabela ANGAJAT, la inserare, ridica eroare daca nu a trecut o saptamana de la ultima angajare, la update, ridica eroare daca incerci sa faci update in timpul weekend-ului iar la stergere ridica eroare daca numarul actual de angajati este unul par.

➤ Cod sursă:

```
--10
--trigger declansat la nivel de comanda
-- sa nu poti inregistra un angajat nou daca
--nu a trecut o saptamana de la ultima angajare
--nu poti sa dai update daca este weekend
--si nu poti sa il stergi decat daca e un numar par de angajati

--la nivelul bazei mele de date, nu sunt cele mai logice triggere,
--insa am incercat sa le fac interesante
/
CREATE OR REPLACE TRIGGER EX10
    BEFORE INSERT OR UPDATE OR DELETE ON ANGAJAT
DECLARE
    VCOUNT NUMBER(4);
BEGIN
    SELECT SYSDATE - MAX(DATA_ANGAJARE) --cate zile de la ultima angajare
    INTO VCOUNT
    FROM ANGAJAT;

    IF INSERTING THEN
        IF (VCOUNT < 7) THEN
            RAISE_APPLICATION_ERROR(-20007,'Nu a trecut inca o saptamana de
la ultima angajare');
        END IF;
    END IF;

    IF UPDATING THEN
        IF(TO_CHAR(SYSDATE,'D') = 6) OR (TO_CHAR(SYSDATE,'D') = 7) THEN
            RAISE_APPLICATION_ERROR(-20008,'Nu puteti face update la angajati
in timpul weekend-ului');
        END IF;
    END IF;

    SELECT COUNT(*)
    INTO VCOUNT
    FROM ANGAJAT;

    IF DELETING THEN
        IF (MOD(VCOUNT,2) = 0) THEN
            RAISE_APPLICATION_ERROR(-20009,'Nu puteti sterge un angajat daca
este numar par de angajati');
        END IF;
    END IF;
```

```
END IF;

END;
/

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,
DATA_ANGAJARE)
VALUES (7, 'Jack Napier', 7777, 'Publicitar', DATE '2025-01-04'); --cineva
angajat acum mai putin de o saptamana

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT,
DATA_ANGAJARE)
VALUES (6, 'Alfred Pennyworth', 10500, 'Management', DATE '1989-02-28'); -- sa
nu il putem inregistra pe el,
--indiferent de data lui de angajare haha

UPDATE ANGAJAT
SET SALARIU = '10000'
WHERE ID_ANGAJAT = 25; --azi, cand pun acest exercitiu in proiect, este
sambata!

DELETE FROM ANGAJAT
WHERE ID_ANGAJAT = 22; --momentan sunt 28 de angajati in baza de date, deci
ar trebui sa se declanseze
--(am drop-at trigger-ul ca sa il adaug pe alfred ca
sa am numar par)

select * from angajat;

/
DROP TRIGGER EX10;
```

➤ Printscrean

The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code area contains the SQL script for creating the trigger EX10. The trigger is defined as follows:

```
9 /
10 CREATE OR REPLACE TRIGGER EX10
11   BEFORE INSERT OR UPDATE OR DELETE ON ANGAJAT
12   DECLARE
13     VCOUNT NUMBER(4);
14 BEGIN
15   SELECT SYSDATE - MAX(DATA_ANGAJARE) --cate zile de la ultima angajare
16   INTO VCOUNT
17   FROM ANGAJAT;
18
19 IF INSERTING THEN
20   IF (VCOUNT < 7) THEN
21     RAISE_APPLICATION_ERROR(-20007, 'Nu a trecut inca o saptamana de la ultima angajare');
22   END IF;
23 END IF;
24
25 IF UPDATING THEN
26   IF(TO_CHAR(SYSDATE,'D') = 6) OR (TO_CHAR(SYSDATE,'D') = 7) THEN
27     RAISE_APPLICATION_ERROR(-20008, 'Nu puteti face update la angajati in timpul weekend-ului');
28   END IF;
29 END IF;
30
31 SELECT COUNT(*)
32   INTO VCOUNT
33   FROM ANGAJAT;
34
35 IF DELETING THEN
36   IF (MOD(VCOUNT,2) = 0) THEN
37     RAISE_APPLICATION_ERROR(-20009, 'Nu puteti sterge un angajat daca este numar par de angajati');
38   END IF;
39 END IF;
40
41 END;
```

The status bar at the bottom indicates "Trigger EX10 compiled".

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (7, 'Jack Napier', 7777, 'Publicitar', DATE '2025-01-04'); --cineva angajat acum mai putin de o saptamana

INSERT INTO ANGAJAT (ID_CASA_DISCURI, NUME, SALARIU, DEPARTAMENT, DATA_ANGAJARE)
VALUES (6, 'Alfred Pennyworth', 10500, 'Management', DATE '1989-02-28'); -- sa nu il putem inregistra pe el,
--indiferent de data lui de angajare haha
```

Script Output | Query Result | Task completed in 0.094 seconds  
Error at Command Line : 495 Column : 8  
Error report -  
SQL Error: ORA-20007: Nu a trecut inca o saptamana de la ultima angajare  
ORA-06512: at "LUCA.EX10", line 10  
ORA-04088: error during execution of trigger 'LUCA.EX10'

```
52
53 UPDATE ANGAJAT
54   SET SALARIU = '10000'
55 WHERE ID_ANGAJAT = 25; --azi, cand pun acest exercitiu in proiect, este sambata!
56
57
```

Script Output | Query Result | Task completed in 0.049 seconds  
Error at Command Line : 53 Column : 12  
Error report -  
SQL Error: ORA-20008: Nu puteti face update la angajati in timpul weekend-ului  
ORA-06512: at "LUCA.EX10", line 16  
ORA-04088: error during execution of trigger 'LUCA.EX10'

```
DELETE FROM ANGAJAT
WHERE ID_ANGAJAT = 22; --momentan sunt 28 de angajati in baza de date, deci ar trebui sa se declanseze
--(am drop-at trigger-ul ca sa il adaug pe alfred ca sa am numar par)
```

Script Output | Query Result | Task completed in 0.056 seconds  
Error at Command Line : 503 Column : 13  
Error report -  
SQL Error: ORA-20009: Nu puteti sterge un angajat daca este numar par de angajati  
ORA-06512: at "LUCA.EX10", line 26  
ORA-04088: error during execution of trigger 'LUCA.EX10'

```
select * from angajat;
```

Script Output | Query Result | All Rows Fetched: 28 in 0.004 seconds

ID_ANGAJAT	ID_CASA_DISCURI	NUME	SALARIU	DEPARTAMENT	DATA_ANGAJARE
25	3	Bruce Wayne	52000	Financiar	20-FEB-22
26	26	Pamela Isley	8700	Financiar	28-FEB-01
27	27	Jack Napier	7777	Publicitar	04-JAN-25
28	28	Alfred Pennyworth	10500	Management	28-FEB-89

---

**11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.**

---

➤ **Cerință:**

Definiți un trigger de tip LMD la nivel de linie pentru tabelul muzician, care, la inserare sa ridice eroare daca muzicianul are varsta sub 16 ani, iar la orice fel de update, sa se extinda contractul activ al muzicianului cu 30 de zile.

➤ Cod sursă:

```
--11
--trigger declansat la nivel de linie
--cand se face update la un muzician se modifica orice contract
--activ al lui, i se mai adauga 30 de zile la acesta
--nu poti sa bagi un muzician care are varsta mai mica de 16 ani
/
CREATE OR REPLACE PROCEDURE EXTINDERE_CONTRACTE
    (V_ID ANGAJAT.ID_ANGAJAT%TYPE) AS
    OK NUMBER(1);
BEGIN
    UPDATE CONTRACT
        SET DATA_EXPIRARE = DATA_EXPIRARE + 30
        WHERE ID_MUZICIAN = V_ID
        AND SYSDATE - DATA_EXPIRARE < 0;
END;
/

CREATE OR REPLACE TRIGGER EX11
    BEFORE INSERT OR UPDATE ON MUZICIAN
    FOR EACH ROW
BEGIN
    IF UPDATING THEN
        EXTINDERE_CONTRACTE(:old.id_muzician);
    END IF;

    IF INSERTING THEN
        IF ( MONTHS_BETWEEN(SYSDATE, :NEW.ZI_NASTERE)/12 < 16) THEN
            RAISE_APPLICATION_ERROR (-20016, 'Nu puteti inregistra un
muzician sub varsta de 16 ani');
        END IF;
    END IF;
END;
/
-----
/
--o sa inserez un muzician nou cu 2 contracte activ si unul inactiv
INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Joker', DATE '1969-08-01', 'm'); --17

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (17, 7, DATE '2023-08-31', DATE '2026-08-31');

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

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
DATA_EXPIRARE)
VALUES (17, 7, DATE '2023-08-31', DATE '2027-02-07');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE,
DATA_EXPIRARE)
VALUES (17, 7, DATE '2023-08-31', DATE '2024-02-07');

SELECT * FROM CONTRACT WHERE ID_MUZICIAN = 17;

UPDATE MUZICIAN
SET NUME = 'The Joker'
WHERE ID_MUZICIAN = 17;

SELECT * FROM CONTRACT WHERE ID_MUZICIAN = 17;

--insearea unui muzician sub varsta de 16 ani
INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Nightwing', DATE '2010-10-10', 'm');

DROP TRIGGER EX11;
```

➤ Printscrean

The screenshot shows the SQL editor in Oracle SQL Developer with the following code:

```
1 --11
2 --trigger declansat la nivel de linie
3 --cand se face update la un muzician se modifica orice contract
4 --activ, i se mai adauga un an la acesta
5 --nu poti sa bagi un muzician care are varsta mai mica de 16 ani
6 /
7 CREATE OR REPLACE PROCEDURE EXTINDERE_CONTRACTE
8     (V_ID ANGAJAT.ID_ANGAJAT%TYPE) AS
9     OK NUMBER(1);
10 BEGIN
11     UPDATE CONTRACT
12         SET DATA_EXPIRARE = DATA_EXPIRARE + 30
13     WHERE ID_MUZICIAN = V_ID
14     AND SYSDATE - DATA_EXPIRARE < 0;
15 END;
16 /
17
18 CREATE OR REPLACE TRIGGER EX11
19     BEFORE INSERT OR UPDATE ON MUZICIAN
20     FOR EACH ROW
21 BEGIN
22     IF UPDATING THEN
23         EXTINDERE_CONTRACTE(:old.id_muzician);
24     END IF;
25
26     IF INSERTING THEN
27         IF ( MONTHS_BETWEEN(SYSDATE,:NEW.ZI_NASTERE)/12 < 16) THEN
28             RAISE_APPLICATION_ERROR (-20016, 'Nu puteti inregistra un muzician sub varsta de 16 ani');
29         END IF;
30     END IF;
31 END;
32 /
```

Below the code, the output window shows:

Procedure EXTINDERE\_CONTRACTE compiled  
Trigger EX11 compiled

--INAINTE DE UPDATE - 2 contracte active si unul expirat

The screenshot shows the SQL editor with the following code:

```
--o sa inserez un muzician nou cu 2 contracte activi si unul inactiv
INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Joker',DATE '1969-08-01', 'm'); --17

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (17,7,DATE'2023-08-31',DATE'2026-08-31');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (17,7,DATE'2023-08-31',DATE'2027-02-07');

INSERT INTO CONTRACT(ID_MUZICIAN, ID_CASA_DISCURI, DATA_SEMNARE, DATA_EXPIRARE)
VALUES (17,7,DATE'2023-08-31',DATE'2024-02-07');

SELECT * FROM CONTRACT WHERE ID_MUZICIAN = 17;
```

Below the code, the output window shows:

All Rows Fetched: 3 in 0.007 seconds

ID_CONTRACT	ID_MUZICIAN	ID_CASA_DISCURI	DATA_SEMNARE	DATA_EXPIRARE
1	16	17	7 31-AUG-23	31-AUG-26
2	17	17	7 31-AUG-23	07-FEB-27
3	18	17	7 31-AUG-23	07-FEB-24

Serba Ioana-Raluca  
Seria 24 Grupa 241

--DUPA UPDATE – cele active s-au extins cu o luna

1 row updated.

```
UPDATE MUZICIAN
SET NUME = 'The Joker'
WHERE ID_MUZICIAN = 17;

SELECT * FROM CONTRACT WHERE ID_MUZICIAN = 17;
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.001 seconds

ID_CONTRACT	ID_MUZICIAN	ID_CASA_DISCURI	DATA_SEMNARE	DATA_EXPIRARE
1	16	17	7 31-AUG-23	30-SEP-26
2	17	17	7 31-AUG-23	09-MAR-27
3	18	17	7 31-AUG-23	07-FEB-24

```
SELECT * FROM CONTRACT WHERE ID_MUZICIAN = 17;

--insearea unui muzician sub varsta de 16 ani
INSERT INTO MUZICIAN (NUME, ZI_NASTERE, GEN)
VALUES ('Nightwing', DATE '2010-10-10', 'm');
```

Script Output x Query Result x

Task completed in 0.057 seconds

```
Error at Command Line : 571 Column : 8
Error report -
SQL Error: ORA-20016: Nu puteti inregistra un muzician sub varsta de 16 ani
ORA-06512: at "LUCA.EX11", line 8
ORA-04088: error during execution of trigger 'LUCA.EX11'
```

---

**12. Definiți un trigger de tip LDD. Declanșați trigger-ul.**

---

➤ **Cerință:**

Cand se efectueaza orice comanda de tip LDD asupra schemei, sa se retine timestamp-ul, numele utilizatorului, actiunea facuta si numele obiectului afectat in tabela ARHIVA.

➤ Cod sursă:

```
--12
--cand se efectueaza orice comanda LDD asupra schemei
--se retine in tabelul ARHIVA
--data, numele utilizatorului, actiunea facuta, tipul si numele tabelului
/
CREATE TABLE ARHIVA
(   DATAE TIMESTAMP(3),
    NUME_USER VARCHAR2(50),
    EVENT VARCHAR2(50),
    TIP_OBIECT VARCHAR2(50),
    NUME_OBIECT VARCHAR2(50));

/
CREATE OR REPLACE TRIGGER EX12
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO ARHIVA
    VALUES(SYSTIMESTAMP, SYS.LOGIN_USER, SYS.SYSEVENT,
           SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME);
END;
/
CREATE TABLE CD (NUME VARCHAR2(40), ID_ALBUM INT);
ALTER TABLE CD
ADD CONSTRAINT FK_ALBUM FOREIGN KEY (ID_ALBUM)
REFERENCES ALBUM (ID_ALBUM);
DROP TABLE CD;

SELECT * FROM ARHIVA;
DROP TABLE ARHIVA;

DROP TRIGGER EX12;

/
```

➤ Printscrean

```
5 /6
577 --12
578 --cand se efectueaza orice comanda LDD asupra schemei
579 --se retine in tabelul ARHIVA
580 --data, numele utilizatorului, actiunea facuta, tipul si numele tabelului
581 /
582 CREATE TABLE ARHIVA
583 (    DATAE TIMESTAMP(3),
584     NUME_USER VARCHAR2(50),
585     EVENT VARCHAR2(50),
586     TIP_OBIECT VARCHAR2(50),
587     NUME_OBIECT VARCHAR2(50));
588 /
589 /
590 CREATE OR REPLACE TRIGGER EX12
591     AFTER CREATE OR DROP OR ALTER ON SCHEMA
592 BEGIN
593     INSERT INTO ARHIVA
594         VALUES(SYSTIMESTAMP, SYS.LOGIN_USER, SYS.SYSEVENT,
595                 SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME);
596 END;
597 /
598 CREATE TABLE CD (NUME VARCHAR2(40), ID_ALBUM INT);
599 ALTER TABLE CD
600 ADD CONSTRAINT FK_ALBUM FOREIGN KEY (ID_ALBUM)
601 REFERENCES ALBUM (ID_ALBUM);
602 DROP TABLE CD;
603
604 SELECT * FROM ARHIVA;
605 DROP TABLE ARHIVA;
606
```

Script Output | Query Result | SQL | All Rows Fetched: 7 in 0.002 seconds

DATAE	NUME_USER	EVENT	TIP_OBIECT	NUME_OBIECT
1 08-JAN-25 20.26.54.433000000	LUCA	DROP TABLE		CD
2 08-JAN-25 20.26.56.819000000	LUCA	CREATE TABLE		CD
3 08-JAN-25 20.26.58.702000000	LUCA	ALTER TABLE		CD
4 08-JAN-25 20.29.12.963000000	LUCA	CREATE PROCEDURE		EXTINDERE_CONTRACTE
5 08-JAN-25 20.29.15.688000000	LUCA	CREATE TRIGGER		EX11
6 08-JAN-25 20.29.20.576000000	LUCA	CREATE PACKAGE		EX13
7 08-JAN-25 20.29.22.933000000	LUCA	CREATE PACKAGE BODY		EX13

(in caz ca ati ajuns in acest punct si va intrebati cine este Luca, este o porecla care vine de la Raluca)

---

### 13. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un pachet

---

care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

➤ **Cerință:**

Definiți un pachet care să pună la dispozitie publica proceduri care afisează informații despre o casă de discuri, un album sau un cântec, fie după ID fie după nume. Sa se definească și 2 proceduri care, primind numele unui album sau unui cântec returnează informații, în special ID-ul, tuturor albumelor sau cântecelor cu acel nume.

Pentru casă de discuri, să se afiseze numele obligatoriu, și restul atributelor dacă nu sunt nule, numărul de departamente, numele acestora și sub fiecare, salariul maxim alături de o listă cu toți angajații și salariul lor.

Pentru album, numele sau și numele trupei care l-a lansat, obligatoriu, tipul și data de lansare dacă există, alături de toate trupele care au contribuit la orice cântec de pe album, în afară de trupa care l-a lansat, împreună cu o listă cu toate cantecele și lungimile lor de pe album.

Pentru cântec, afișați numele și lungimea, numele tuturor muzicienilor care au contribuit la cântec, alături de poziția lor în trupa.

➤ Cod sursă:

```
--13
--pachet
--tipuri de date complexe = vector in vector / record

--definiți un pachet care să facă disponibil public proceduri care returnează
detalii despre
--casa de discuri, albumul sau cântecul
--al cărui ID sau nume este dat

CREATE OR REPLACE PACKAGE EX13 AS
--DECLARAȚII PUBLICE
--2 proceduri pentru fiecare: casa discuri, album și cântec
--una cu numele, una cu ID-ul
--la album și cântec, exceptii dacă există mai multe cu același nume (daca se
apelează prin nume)

    PROCEDURE ABOUT_CASA_DISCURI (CD_ID IN
CASA_DISCURI.ID_CASA_DISCURI%TYPE);
    PROCEDURE ABOUT_CASA_DISCURI (CD_NUME IN CASA_DISCURI.NUME%TYPE);

    PROCEDURE ABOUT_ALBUM (VALBUM_ID IN ALBUM.ID_ALBUM%TYPE);
    PROCEDURE ABOUT_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE);

    PROCEDURE ABOUT_CANTEC (VCANTEC_ID IN CANTEC.ID_CANTEC%TYPE);
    PROCEDURE ABOUT_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE);

    PROCEDURE GASESTE_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE);
    PROCEDURE GASESTE_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE);

END EX13;
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
/CREATE OR REPLACE PACKAGE BODY EX13 IS
--DECLARATII PRIVATE
  TYPE T_ID_TRUPE_CANTEC IS TABLE OF TRUPA.ID_TRUPA%TYPE INDEX BY
PLS_INTEGER;

--FUNCTIE CARE RETURNEAZA ID-URILE TUTUROR TRUPELOR CARE AU LUCRAT LA UN
CANTEC -PT ALBUM

FUNCTION TRUPE_CANTEC (VID_CANTEC CANTEC.ID_CANTEC%TYPE)
  RETURN T_ID_TRUPE_CANTEC IS
  LISTA_TRUPE T_ID_TRUPE_CANTEC;
BEGIN
  SELECT ID_TRUPA
  BULK COLLECT INTO LISTA_TRUPE
  FROM TRUPA_CANTEC
  WHERE ID_CANTEC = VID_CANTEC;
  RETURN LISTA_TRUPE;
END TRUPE_CANTEC;

--procedurile dupa nume gasesc id-ul sau ridica eroare la album si cantec
daca exista mai multe cu acelasi nume
--si apeleaza procedurile dupa id

--ABOUT CASA DISCURI
  --NUME, ADRESA, WEBSITE, AN INFIINTARE DACA EXISTA
  --NUMARUL DE DEPARTAMENTE, LISTA CU DEPARTAMENTELE
  --SUB FIECARE NUME DE DEPARTAMENT, salariu maxim + LISTA CU TOTI ANGAJATII
SI SALARIU
  --} TABEL(DEPARTAMENT, SALARIUMAX, TABEL DE RECORDS (ANGAJATI + SALARIU))

  --dupa id
PROCEDURE ABOUT_CASA_DISCURI (CD_ID IN CASA_DISCURI.ID_CASA_DISCURI%TYPE)
IS
  --declarari
    NUME_CD CASA_DISCURI.NUME%TYPE;
    ADRESA_CD CASA_DISCURI.ADRESA%TYPE;
    WEBSITE_CD CASA_DISCURI.WEBSITE%TYPE;
    AN_CD CASA_DISCURI.AN_INFIINTARE%TYPE;
    I NUMBER(2);
    VCOUNT NUMBER(1);
    NO_CD EXCEPTION;

    TYPE ANGAJAT_DEPT IS RECORD(
      NUME ANGAJAT.NUME%TYPE,
      SALARIU ANGAJAT.NUME%TYPE); --DETALII ANGAJAT

    TYPE T_ANGAJAT_DEPT IS TABLE OF ANGAJAT_DEPT INDEX BY PLS_INTEGER;

    TYPE R_DETALII_DEPT IS RECORD(
      NUME ANGAJAT.DEPARTAMENT%TYPE,
      SALARIU_MAX ANGAJAT.SALARIU%TYPE,
      DETALII_ANGAJATI T_ANGAJAT_DEPT); --LISTA ANGAJATI PER DEPARTAMENT

    TYPE T_DETALII_DEPT IS TABLE OF R_DETALII_DEPT INDEX BY PLS_INTEGER;
    DETALII_DEPT T_DETALII_DEPT := T_DETALII_DEPT(); --LISTA DEPARTENTE +
DETALII PER CD
```

```
CURSOR DEPARTAMENTE(V_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE) IS
    SELECT UNIQUE DEPARTAMENT
    FROM ANGAJAT
    WHERE V_ID = ID_CASA_DISCURI;

FUNCTION MAXSALARY(VCD CASA_DISCURI.ID_CASA_DISCURI%TYPE, VDEPT
ANGAJAT.DEPARTAMENT%TYPE)
    RETURN ANGAJAT.SALARIU%TYPE IS
    VMAX ANGAJAT.SALARIU%TYPE; --functie pentru salariul maxim al
unui departament al unei cd
BEGIN
    SELECT MAX(SALARIU)
    INTO VMAX
    FROM ANGAJAT
    WHERE DEPARTAMENT = VDEPT
    AND ID_CASA_DISCURI = VCD;
    RETURN VMAX;
END;

BEGIN
    --DACA EXISTA CD
    SELECT COUNT(ID_CASA_DISCURI)
    INTO VCOUNT
    FROM CASA_DISCURI
    WHERE ID_CASA_DISCURI = CD_ID;

    IF VCOUNT = 0 THEN
        RAISE NO_CD;
    END IF;

    --CAUT NUME, SI RESTUL CHESTIILOR DACA EXISTA
    SELECT NUME, ADRESA, WEBSITE, AN_INFIINTARE
    INTO NUME_CD, ADRESA_CD, WEBSITE_CD, AN_CD
    FROM CASA_DISCURI
    WHERE ID_CASA_DISCURI = CD_ID;

    --afisez detaliiile casei de discuri
    DBMS_OUTPUT.PUT_LINE(NUME_CD);
    IF AN_CD IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('An infiintare: ' || AN_CD);
    END IF;
    IF ADRESA_CD IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Adresa: ' || ADRESA_CD);
    END IF;
    IF WEBSITE_CD IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Website: ' || WEBSITE_CD);
    END IF;
    DBMS_OUTPUT.PUT_LINE('-----');

    --TREC PRIN DEPARTAMENTE
    --POPULEZ PENTRU FIECARE TABELUL DE ANGAJATI
    I := 0;
    FOR VDEPT IN DEPARTAMENTE(CD_ID) LOOP
        I := I + 1;
```

```
DETALII_DEPT(I).NUME := VDEPT.DEPARTAMENT;
DETALII_DEPT(I).SALARIU_MAX := MAXSALARY(CD_ID,
VDEPT.DEPARTAMENT);

SELECT NUME, SALARIU
BULK COLLECT INTO DETALII_DEPT(I).DETALII_ANGAJATI
FROM ANGAJAT
WHERE DEPARTAMENT = VDEPT.DEPARTAMENT
AND ID_CASA_DISCURI = CD_ID;
END LOOP;

--afisam departamentele si angajatii, daca exista
IF DETALII_DEPT.COUNT > 0 THEN
    DBMS_OUTPUT.PUT_LINE('Nr. departamente: ' || DETALII_DEPT.COUNT);
    FOR J IN DETALII_DEPT.FIRST..DETALII_DEPT.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('-----');
        DBMS_OUTPUT.PUT_LINE(DETALII_DEPT(J).NUME);
        DBMS_OUTPUT.PUT_LINE('---');
        DBMS_OUTPUT.PUT_LINE('Salariul maxim:');
        ' || DETALII_DEPT(J).SALARIU_MAX);
        --afisam toti angajatii (există obligatoriu minim 1)
        FOR K IN
DETALII_DEPT(J).DETALII_ANGAJATI.FIRST..DETALII_DEPT(J).DETALII_ANGAJATI.LAST
LOOP

DBMS_OUTPUT.PUT_LINE(DETALII_DEPT(J).DETALII_ANGAJATI(K).NUME
|| ' ' || 
DETALII_DEPT(J).DETALII_ANGAJATI(K).SALARIU);
        END LOOP;
    END LOOP;
ELSE
    DBMS_OUTPUT.PUT_LINE('Casa de discuri nu are departamente sau
angajati');
END IF;
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.NEW_LINE;
EXCEPTION
    WHEN NO_CD THEN RAISE_APPLICATION_ERROR(-20001,'Nu există casa de
discuri cu acel id');
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20001,'nu au fost
gasite date');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost
returnate prea multe linii');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20001,'alta eroare');
END ABOUT_CASA_DISCURI;

--după nume
PROCEDURE ABOUT_CASA_DISCURI (CD_NUME IN CASA_DISCURI.NUME%TYPE) IS
    CD_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE;
    VCOUNT NUMBER(1);
    NO_CD EXCEPTION;
BEGIN
    --DACA EXISTA
    SELECT COUNT(ID_CASA_DISCURI)
    INTO VCOUNT
    FROM CASA_DISCURI
    WHERE UPPER(CASA_DISCURI.NUME) = UPPER(CD_NUME);
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
--OBLIGATORIU EXISTA MAXIM 1 CD CU ACEL NUME
IF VCOUNT>0 THEN
    SELECT ID_CASA_DISCURI
    INTO CD_ID
    FROM CASA_DISCURI
    WHERE UPPER(CASA_DISCURI.NUME) = UPPER(CD_NUME);
    ABOUT_CASA_DISCURI(CD_ID);
ELSE
    RAISE NO_CD;
END IF;

EXCEPTION
    WHEN NO_CD THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista casa de
discuri cu acel nume');
END ABOUT_CASA_DISCURI;

--ABOUT ALBUM
--NUME, NUMELE TRUPEI CARE A LANSAT, TIPUL SI DATA LANSARII DACA EXISTA
--CE TRUPE AU MAI CONTRIBUIT FOLOSIND FUNCTIA = TABEL INDEXAT, INDEX =
ID_TRUPA
--LUNGIME TOTALA
--TOATE CANTECLE = TABEL INDEXAT

--dupa id
PROCEDURE ABOUT_ALBUM (VALBUM_ID IN ALBUM.ID_ALBUM%TYPE) IS
--declaratii
    NUME_ALBUM ALBUM.NUME%TYPE;
    NUME_TRUPA TRUPA.NUME%TYPE;
    TIP_ALBUM ALBUM.TIP%TYPE;
    DATA_ALBUM ALBUM.DATA_LANSARE%TYPE;
    DURATA_ALBUM CANTEC.LUNGIME%TYPE;

    LISTA_ID_TRUPE T_ID_TRUPE_CANTEC;
    TEMP_NUME TRUPA.NUME%TYPE;

    TYPE T_FEAT_TRUPE IS TABLE OF NUMBER INDEX BY TRUPA.NUME%TYPE;
    FEAT_TRUPE T_FEAT_TRUPE; --NUMELE TRUPELOR CARE AU CONTRIBUIT,
INDEXATE DUPA NUMELE TRUPEI
    --CA SA FIE UNICE
    INDX TRUPA.NUME%TYPE;

    TYPE DETALII_CANTEC IS RECORD(
        NUME CANTEC.NUME%TYPE,
        LUNGIME CANTEC.LUNGIME%TYPE);

    TYPE T_CANTECE_ALBUM IS TABLE OF DETALII_CANTEC INDEX BY PLS_INTEGER;
    CANTECE_ALBUM T_CANTECE_ALBUM;

    VCOUNT NUMBER(2);
    NO_ALBUM EXCEPTION;

    CURSOR CANTECE(VID_ALBUM ALBUM.ID_ALBUM%TYPE) IS
        SELECT ID_CANTEC
        FROM CANTEC_ALBUM
        WHERE ID_ALBUM = VID_ALBUM;
```

```
BEGIN
    --DACA EXISTA ALBUMUL
    SELECT COUNT(ID_ALBUM)
    INTO VCOUNT
    FROM ALBUM
    WHERE ID_ALBUM = VALBUM_ID;

    IF VCOUNT = 0 THEN
        RAISE NO_ALBUM;
    END IF;

    --CAUT NUME, SI RESTUL CHESTIILOR DACA EXISTA
    SELECT A.NUME, T.NUME, TIP, DATA_LANSARE
    INTO NUME_ALBUM, NUME_TRUPA, TIP_ALBUM, DATA_ALBUM
    FROM ALBUM A
    JOIN TRUPA T ON A.ID_TRUPA = T.ID_TRUPA
    WHERE ID_ALBUM = VALBUM_ID;

    --aflam numele trupelor care au contribuit la cantece de pe album,
    daca exista
    --trec prin toate cantecele, si folosind functia retain in tabel
    numele trupelor

    FOR VCANTEC IN CANTECE(VALBUM_ID) LOOP
        LISTA_ID_TRUPE := TRUPE_CANTEC(VCANTEC.ID_CANTEC);
        IF LISTA_ID_TRUPE.COUNT > 0 THEN --in cazul in care albumul nu
        are cantece inregistrate sau cantecele nu au trupe inregistrate
            FOR I IN LISTA_ID_TRUPE.FIRST..LISTA_ID_TRUPE.LAST LOOP
                SELECT NUME
                INTO TEMP_NUME
                FROM TRUPA
                WHERE ID_TRUPA = LISTA_ID_TRUPE(I);
                IF TEMP_NUME != NUME_TRUPA THEN
                    FEAT_TRUPE(TEMP_NUME) := 1; --PRACTIC, IMI PASA DOAR
                END IF;
            END LOOP;
        END IF;
    END LOOP;

    --si populez si tabelul de cantece
    SELECT C.NUME, C.LUNGIME
    BULK COLLECT INTO CANTECE_ALBUM
    FROM CANTEC C
    JOIN CANTEC_ALBUM CA ON C.ID_CANTEC = CA.ID_CANTEC
    WHERE ID_ALBUM = VALBUM_ID;

    --aflam durata totala
    SELECT SUM(LUNGIME)
    INTO DURATA_ALBUM
    FROM CANTEC C
    JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
    WHERE CA.ID_ALBUM = VALBUM_ID;
```

```
--AFISAM detaliiile albumului
DBMS_OUTPUT.PUT_LINE(NUME_ALBUM);
IF TIP_ALBUM IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Tip: ' || TIP_ALBUM);
END IF;
DBMS_OUTPUT.PUT_LINE('Lansat de: ' || NUME_TRUPA);

--FEAT TRUPELE CARE AU CONTRIBUIT
IF FEAT_TRUPE.COUNT > 0 THEN
    INDX := FEAT_TRUPE.FIRST;
    DBMS_OUTPUT.PUT_LINE('Featuring: ' || INDX);
    INDX := FEAT_TRUPE.NEXT(INDX);
    WHILE INDX IS NOT NULL LOOP
        DBMS_OUTPUT.PUT_LINE(''           ' || INDX);
        INDX := FEAT_TRUPE.NEXT(INDX);
    END LOOP;
END IF;
IF DATA_ALBUM IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Data lansarii: ' || DATA_ALBUM);
END IF;
DBMS_OUTPUT.PUT_LINE('Durata: ' || NVL(DURATA_ALBUM, 0) || ' min');
DBMS_OUTPUT.PUT_LINE('-----');

IF CANTECE_ALBUM.COUNT > 0 THEN
    FOR I IN CANTECE_ALBUM.FIRST..CANTECE_ALBUM.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(I || ' - '
        || TO_CHAR(CANTECE_ALBUM(I).LUNGIME, 'FM990.00') || ' - '
        || CANTECE_ALBUM(I).NUME); --+ FORMATARE sa fie mereu 2
zecimale
    END LOOP;
ELSE
    DBMS_OUTPUT.PUT_LINE('Albumul nu are cantece inregistrate');
END IF;
DBMS_OUTPUT.NEW_LINE;

EXCEPTION
    WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20002, 'Nu exista albumul
cu acel id');
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20002, 'nu au fost
gasite date');
    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20002, 'au fost
returnate prea multe linii');
    WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002, 'alta eroare');
END ABOUT_ALBUM;

-- DUPA NUME
PROCEDURE ABOUT_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE) IS
VCOUNT NUMBER(2);
ALBUM_ID ALBUM.ID_ALBUM%TYPE;
TOO_MANY_ALBUM EXCEPTION;
NO_ALBUM EXCEPTION;
BEGIN
    --DACA EXISTA
    SELECT COUNT(ID_ALBUM)
    INTO VCOUNT
    FROM ALBUM
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);

IF VCOUNT > 1 THEN
    RAISE TOO_MANY_ALBUM;
END IF;

IF VCOUNT = 1 THEN
    SELECT ID_ALBUM
    INTO ALBUM_ID
    FROM ALBUM
    WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);
    ABOUT_ALBUM(ALBUM_ID);
ELSE
    RAISE NO_ALBUM;
END IF;
EXCEPTION
    WHEN TOO_MANY_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Exista mai
multe albume cu acel nume. Folositi GASESTE_ALBUM pentru a gasi id-ul
albumului pe care il doriti');
    WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Nu exista albume
cu acel nume');
END ABOUT_ALBUM;

--ABOUT CANTEC
--NUME, LUNGIME
--NUMELE MUZICIENILOR (NU TRUPE!) CARE AU CONTRIBUIT LA CANTEC
-- + POZITIA IN TRUPA } TABEL(ID_TRUPA) CU RECORD(NUME,POZITIE)) -FOLOSIND
FUNCTIONIA

--DUPA ID
PROCEDURE ABOUT_CANTEC (VCANTEC_ID IN CANTEC.ID_CANTEC%TYPE) IS
    NUME_CANTEC CANTEC.NUME%TYPE;
    LUNGIME_CANTEC CANTEC.LUNGIME%TYPE;
    VCOUNT NUMBER(2);
    NO_CANTEC EXCEPTION;

    TYPE R_DETALII_MUZICIENI IS RECORD(
        NUME MUZICIAN.NUME%TYPE,
        POZITIE MEMBRU_TRUPA.POZITIE%TYPE);
    TYPE T_DETALII_MUZICIENI IS TABLE OF R_DETALII_MUZICIENI INDEX BY
PLS_INTEGER;

    TYPE R_TRUPA_MUZICIENI IS RECORD(
        TRUPA_ID TRUPA.ID_TRUPA%TYPE,
        MUZICIENI T_DETALII_MUZICIENI); --pt fiecare id trupa, numele si
pozitia muzicienilor
    TYPE T_TRUPA_MUZICIENI IS TABLE OF R_TRUPA_MUZICIENI INDEX BY
PLS_INTEGER;

    MUZICIENI_CANTEC T_TRUPA_MUZICIENI;

    -- FUNCTIE CARE RETURNEAZA MUZICIENII UNUI CANTEC- SI FOLOSESTE
TRUPE_CANTEC
    FUNCTION F_MUZICIENI_CANTEC (VID_CANTEC CANTEC.ID_CANTEC%TYPE)
        RETURN T_TRUPA_MUZICIENI IS
        TRUPA_MUZICIENI T_TRUPA_MUZICIENI;
        ID_TRUPE_CANTEC T_ID_TRUPE_CANTEC;
```

```
BEGIN
    ID_TRUPE_CANTEC := TRUPE_CANTEC(VID_CANTEC); --aici am id-urile
    trupelor care au contribuit la acest cantec
    IF ID_TRUPE_CANTEC.COUNT>0 THEN --in caz ca cantecul nu are trupe
        inregistrate
            FOR I IN ID_TRUPE_CANTEC.FIRST..ID_TRUPE_CANTEC.LAST LOOP
                TRUPA_MUZICIENI(I).TRUPA_ID := ID_TRUPE_CANTEC(I); --
                PENTRU FIECARE ID DE TRUPA
                    SELECT M.NUME, MT.POZITIE --IMI PUN TOTI MUZICIENII
                    TRUPEI
                        BULK COLLECT INTO TRUPA_MUZICIENI(I).MUZICIENI
                        FROM MUZICIAN M
                        JOIN MEMBRU_TRUPA MT ON M.ID_MUZICIAN = MT.ID_MUZICIAN
                        WHERE MT.ID_TRUPA = ID_TRUPE_CANTEC(I);
                    END LOOP;
            END IF;
            RETURN TRUPA_MUZICIENI;
    END F_MUZICIENI_CANTEC;

BEGIN
    --DACA EXISTA CANTECTUL
    SELECT COUNT(ID_CANTEC)
    INTO VCOUNT
    FROM CANTEC
    WHERE ID_CANTEC = VCANTEC_ID;

    IF VCOUNT = 0 THEN
        RAISE NO_CANTEC;
    END IF;

    --CAUT NUME, SI RESTUL CHESTIILOR DACA EXISTA
    SELECT NUME, LUNGIME
    INTO NUME_CANTEC, LUNGIME_CANTEC
    FROM CANTEC C
    WHERE ID_CANTEC = VCANTEC_ID;

    --POPULEZ TABELUL CU DETALIILE MUZICIENILOR
    MUZICIENI_CANTEC := F_MUZICIENI_CANTEC(VCANTEC_ID);

    --AFISEZ FRUMOS
    DBMS_OUTPUT.PUT_LINE(NUME_CANTEC);
    DBMS_OUTPUT.PUT_LINE('Durata:');
    ' || TO_CHAR(LUNGIME_CANTEC,'FM9990.00') || ' min');
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('Credits:');
    IF MUZICIENI_CANTEC.COUNT > 0 THEN
        FOR I IN MUZICIENI_CANTEC.FIRST..MUZICIENI_CANTEC.LAST LOOP
            FOR J IN
                MUZICIENI_CANTEC(I).MUZICIENI.FIRST..MUZICIENI_CANTEC(I).MUZICIENI.LAST LOOP
                    DBMS_OUTPUT.PUT_LINE(MUZICIENI_CANTEC(I).MUZICIENI(J).NUME || ' - ' ||
                        MUZICIENI_CANTEC(I).MUZICIENI(J).POZITIE );
                END LOOP;
        END LOOP;
    ELSE DBMS_OUTPUT.PUT_LINE('Cantecul nu are muzicieni inregistrati');
    END IF;
```

```
    EXCEPTION
        WHEN NO_CANTEC THEN RAISE_APPLICATION_ERROR(-20003,'Nu exista cantece cu acel id');
        WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20003,'nu au fost gasite date');
        WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20003,'au fost returnate prea multe linii');
        WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20003,'alta eroare');

    END ABOUT_CANTEC;

--dupa nume
PROCEDURE ABOUT_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE) IS
    VCOUNT NUMBER(2);
    CANTEC_ID CANTEC.ID_CANTEC%TYPE;
    TOO_MANY_CANTEC EXCEPTION;
    NO_CANTEC EXCEPTION;
    BEGIN
        --DACA EXISTA
        SELECT COUNT(ID_CANTEC)
        INTO VCOUNT
        FROM CANTEC
        WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);

        IF VCOUNT > 1 THEN
            RAISE TOO_MANY_CANTEC;
        END IF;

        IF VCOUNT = 1 THEN
            SELECT ID_CANTEC
            INTO CANTEC_ID
            FROM CANTEC
            WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);
            ABOUT_CANTEC(CANTEC_ID);
        ELSE
            RAISE NO_CANTEC;
        END IF;
    EXCEPTION
        WHEN TOO_MANY_CANTEC THEN RAISE_APPLICATION_ERROR(-20003,'Exista mai multe cantece cu acel nume. Folositi GASESTE_CANTEC pentru a gasi id-ul canteceleui pe care il doriti');
        WHEN NO_CANTEC THEN RAISE_APPLICATION_ERROR(-20003,'Nu exista cantece cu acel nume');
    END ABOUT_CANTEC;

--procedurile de gasit id-uri la album si cantece
PROCEDURE GASESTE_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE) IS
    NO_ALBUM EXCEPTION;
    VCOUNT NUMBER(2);

    CURSOR ALBUME( VNUME IN ALBUM.NUME%TYPE) IS
        SELECT A.ID_ALBUM, A.NUME, T.NUME AS NUME_TRUPA, TIP
        FROM ALBUM A
        JOIN TRUPA T ON A.ID_TRUPA = T.ID_TRUPA
        WHERE UPPER(A.NUME) = UPPER(VNUME);
```

Şerba Ioana-Raluca  
Seria 24 Grupa 241

```
BEGIN
    SELECT COUNT(ID_ALBUM)
    INTO VCOUNT
    FROM ALBUM
    WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);

    IF VCOUNT = 0 THEN
        RAISE NO_ALBUM;
    END IF;

    FOR ALBM IN ALBUME(VALBUM_NUME) LOOP
        DBMS_OUTPUT.PUT_LINE('ID: '||ALBM.ID_ALBUM);
        DBMS_OUTPUT.PUT_LINE('Nume: '||ALBM.NUME);
        DBMS_OUTPUT.PUT_LINE('Nume trupa: '||ALBM.NUME_TRUPA);
        DBMS_OUTPUT.PUT_LINE('Tip: '||ALBM.TIP);
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
    EXCEPTION
        WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20022,'Nu exista
albume cu acel nume');
    END GASESTE_ALBUM;

PROCEDURE GASESTE_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE) IS
    NO_CANTEC EXCEPTION;
    VCOUNT NUMBER(2);

    CURSOR CANTECE( VNUME IN CANTEC.NUME%TYPE) IS
        SELECT ID_CANTEC, NUME, LUNGIME
        FROM CANTEC C
        WHERE UPPER(C.NUME) = UPPER(VNUME);

    CURSOR NUME_TRUPE (VID_CANTEC CANTEC.ID_CANTEC%TYPE) IS
        SELECT NUME
        FROM TRUPA T
        JOIN TRUPA_CANTEC TC ON TC.ID_TRUPA = T.ID_TRUPA
        WHERE TC.ID_CANTEC = VID_CANTEC;

BEGIN
    SELECT COUNT(ID_CANTEC)
    INTO VCOUNT
    FROM CANTEC
    WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);

    IF VCOUNT = 0 THEN
        RAISE NO_CANTEC;
    END IF;

    FOR CT IN CANTECE(VCANTEC_NUME) LOOP
        DBMS_OUTPUT.PUT_LINE('ID: '||CT.ID_CANTEC);
        DBMS_OUTPUT.PUT_LINE('Nume: '||CT.NUME);
        DBMS_OUTPUT.PUT_LINE('Lungime:
'||TO_CHAR(CT.LUNGIME,'FM9990.00')||' min');
        DBMS_OUTPUT.PUT_LINE('Trupe: ');
        FOR CTT IN NUME_TRUPE(CT.ID_CANTEC) LOOP
            DBMS_OUTPUT.PUT_LINE(CTT.NUME);
        END LOOP;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
END;
```

Serba Ioana-Raluca  
Seria 24 Grupa 241

```
        END LOOP;
    EXCEPTION
        WHEN NO_CANTEC THEN RAISE_APPLICATION_ERROR(-20033, 'Nu exista
cantece cu acel nume');
    END GASESTE_CANTEC;

END EX13;
/
BEGIN
    EX13.ABOUT_CANTEC('online');
END;
/
```

## ➤ Printscrean

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Script Output' tab displays the PL/SQL code for package EX13. The code includes several procedures for querying albums and artists, and a function that returns the ID of a specific artist based on their name. The 'Query Result' tab on the right shows the output of the 'GASESTE\_CANTEC' function, which lists department names and their respective maximum salaries. Below the tabs, the status bar indicates the task was completed in 0.041 seconds.

```
Worksheet | Query Builder
10 CREATE OR REPLACE PACKAGE EX13 AS
11 --DECLARATII PUBLICE
12 --2 proceduri pentru fiecare: casa discuri, album si cantic
13 --una cu numele, una cu id-ul
14 --la album si cantic, exceptii daca exista mai multe cu acelasi nume (daca se apeleaza prin nume)
15
16 PROCEDURE ABOUT_CASA_DISCURI (CD_ID IN CASA_DISCURI.ID_CASA_DISCURI%TYPE);
17 PROCEDURE ABOUT_CASA_DISCURI (CD_NUME IN CASA_DISCURI.NUME%TYPE);
18
19 PROCEDURE ABOUT_ALBUM (VALBUM_ID IN ALBUM.ID_ALBUM%TYPE);
20 PROCEDURE ABOUT_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE);
21
22 PROCEDURE ABOUT_CANTEC (VCANTEC_ID IN CANTEC.ID_CANTEC%TYPE);
23 PROCEDURE ABOUT_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE);
24
25 PROCEDURE GASESTE_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE);
26 PROCEDURE GASESTE_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE);
27
28 END EX13;
29 /
30 CREATE OR REPLACE PACKAGE BODY EX13 IS
31 --DECLARATII PRIVATE
32     TYPE T_ID_TRUPE_CANTEC IS TABLE OF TRUPA.ID_TRUPA%TYPE INDEX BY PLS_INTEGER;
33
34 --FUNCTIE CARE RETURNEAZA ID-URILE TUTUROR TRUPELOR CARE AU LUCRAT LA UN CANTEC -PT ALBUM
35
36 FUNCTION TRUPE_CANTEC (VID_CANTEC CANTEC.ID_CANTEC%TYPE)...
37
38 --procedurile dupa nume gasesc id-ul sau ridica eroare la album si cantic daca exista mai multe cu acelasi nume
39 --si apeleaza procedurile dupa id
40
41 --ABOUT CASA DISCURI
42 --NUME, ADRESA, WEBSITE, AN INFIIINTARE DACA EXISTA
43
44
45
46
47
48
49
50
51
```

Script Output | Query Result | Task completed in 0.041 seconds

Package EX13 compiled

Package Body EX13 compiled

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Department	Max Salary	Employee
UNDF	2011	An infiintare: 2011
		Adresa: Melbourne, Victoria
		Website: https://unfdcentral.com
Nr. departamente: 5		
Management		
		Salariul maxim: 6500
		Tucker Rule 6500
Administrativ		
		Salariul maxim: 15800
		Sam Winchester 15800
Marketing		
		Salariul maxim: 7900
		Kai Firre 7900
Audio		
		Salariul maxim: 5500
		Cole Wearth 5500
Financiar		
		Salariul maxim: 9200
		Bruce Wayne 9200
		Pamela Isley 8700

Serba Ioana-Raluca  
Seria 24 Grupa 241

Worksheet | Query Builder

```

50  -->ABOUT CASA DISCURI
51  --NUME, ADRESA, WEBSITE, AN INFINTARE DACA EXISTA
52  --NUMARUL DE DEPARTAMENTE, LISTA CU DEPARTAMENTELA
53  --SUB PIECARE NUME DE DEPARTAMENT, SALARIU MAXIM + LISTA CU TOTI ANGAJATII SI SALARIU
54  --) TABEL(DEPARTAMENT, SALARIUMAX, TABEL DE RECORDS (ANGAJATI + SALARIUI)
55
56
57  --dupa id
58  PROCEDURE ABOUT_CASA_DISCURI (CD_ID IN CASA_DISCURI.ID_CASA_DISCURI%TYPE) IS
59  --declarari
60  NUME_CD CASA_DISCURI.NUME%TYPE;
61  ADRESA_CD CASA_DISCURI.ADRESA%TYPE;
62  WEBSITE_CD CASA_DISCURI.WEB SITE%TYPE;
63  AN_CD CASA_DISCURI.AN_INFINTARE%TYPE;
64  I NUMBER(2);
65  VCOUNT NUMBER(1);
66  NO_CD EXCEPTION;
67
68  TYPE ANGAJAT_DEPT IS RECORD(
69      NUME_ANGAGAT.NUME%TYPE,
70      SALARIU_ANGAGAT.SALARIU%TYPE,
71      SALARIU_ANGAGAT.NUME%TYPE); --DETALII ANGAJAT
72
73  TYPE T_ANGAJAT_DEPT IS TABLE OF ANGAJAT_DEPT INDEX BY PLS_INTEGER;
74
75  TYPE R_DETALII_DEPT IS RECORD(
76      NUME_ANGAGAT.DEPARTAMENT%TYPE,
77      SALARIU_ANGAGAT.SALARIU%TYPE,
78      DETALII_ANGAGAT_T_ANGAGAT_DEPT)--LISTA ANGAJATI PER DEPARTAMENT
79
80  TYPE T_DETALII_DEPT IS TABLE OF R_DETALII_DEPT INDEX BY PLS_INTEGER;
81  DETALII_DEPT T_DETALII_DEPT := T_DETALII_DEPT(); --LISTA DEPARTAMENTE + DETALII PER CD
82
83  CURSOR DEPARTAMENTE(V_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE) IS
84      SELECT UNIQUE DEPARTAMENT
85      FROM ANGAJAT
86      WHERE V_ID = ID_CASA_DISCURI;
87
88  FUNCTION MAXSALARY(VCD CASA_DISCURI.ID_CASA_DISCURI%TYPE, VDEPT ANGAGAT.DEPARTAMENT%TYPE)
89  RETURN ANGAGAT.SALARIU%TYPE IS
90      VMAX ANGAGAT.SALARIU%TYPE; --functie pentru salariul maxim al unui departament al unei cd
91
92  BEGIN
93      SELECT MAX(SALARIU)
94      INTO VMAX
95      FROM ANGAJAT
96      WHERE DEPARTAMENT = VDEPT
97      AND ID_CASA_DISCURI = VCD;
98      RETURN VMAX;
99
100 END;
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135

```

Script Output | Query Result | Task completed in 6.06 seconds  
PL/SQL procedure successfully completed.

Worksheet | Query Builder

```

94      AND ID_CASA_DISCURI = VCD;
95      RETURN VMAX;
96  END;
97
98  BEGIN
99      --DACA EXISTA CD
100     SELECT COUNT(ID_CASA_DISCURI)
101     INTO VCOUNT
102     FROM CASA_DISCURI
103     WHERE ID_CASA_DISCURI = CD_ID;
104
105
106     IF VCOUNT = 0 THEN
107         RAISE NO_CD;
108     END IF;
109
110     --CAUT NUME, SI RESTUL CHESTIILOR DACA EXISTA
111     SELECT* NUME, ADRESA, WEBSITE, AN_INFINTARE
112     INTO NUME_CD, ADRESA_CD, WEBSITE_CD, AN_CD
113     FROM CASA_DISCURI
114     WHERE ID_CASA_DISCURI = CD_ID;
115
116     --afisez detalii casei de discuri
117     DBMS_OUTPUT.PUT_LINE(NUME_CD);
118     IF AN_CD IS NOT NULL THEN
119         DBMS_OUTPUT.PUT_LINE('An infintare: ' || AN_CD);
120     END IF;
121     IF ADRESA_CD IS NOT NULL THEN
122         DBMS_OUTPUT.PUT_LINE('Adresa: ' || ADRESA_CD);
123     END IF;
124     IF WEBSITE_CD IS NOT NULL THEN
125         DBMS_OUTPUT.PUT_LINE('Website: ' || WEBSITE_CD);
126     END IF;
127     DBMS_OUTPUT.PUT_LINE('-----');
128
129
130     --TREC PRIN DEPARTAMENTE
131     --POPULEZ PENTRU PIECARE TABELUL DE ANGAJATI
132     I := 0;
133     FOR VDEPT IN DEPARTAMENTE(CD_ID) LOOP
134         I := I + 1;
135         DETALII_DEPT(I).NUME := VDEPT.DEPARTAMENT;

```

Script Output | Query Result | Task completed in 0.057 seconds  
ERROR at line 1:  
ORA-20001: Nu exista casa de discuri cu acel id  
ORA-06512: at "LUCA.EX13", line 136  
ORA-06512: at line 2  
<https://docs.oracle.com/error-help/db/ora-20001/>

Serba Ioana-Raluca  
Seria 24 Grupa 241

Worksheet    Query Builder

```
135 DETALII_DEPT(I).NUME := VDEPT.DEPARTAMENT;
136 DETALII_DEPT(I).SALARIU_MAX := MAXSALARY(CD_ID, VDEPT.DEPARTAMENT);
137
138   SELECT NUME, SALARIU
139   BULK COLLECT INTO DETALII_DEPT(I).DETALII_ANGAJATI
140   FROM ANGAJAT
141   WHERE DEPARTAMENT = VDEPT.DEPARTAMENT
142   AND ID_CASA_DISCURI = CD_ID;
143   END LOOP;
144
145 --afisam departamentele si angajatii, daca exista
146 IF DETALII_DEPT.COUNT > 0 THEN
147   DBMS_OUTPUT.PUT_LINE('Nr. departamente: ' || DETALII_DEPT.COUNT);
148   FOR J IN DETALII_DEPT.FIRST..DETALII_DEPT.LAST LOOP
149     DBMS_OUTPUT.PUT_LINE('-----');
150     DBMS_OUTPUT.PUT_LINE(DETALII_DEPT(J).NUME);
151     DBMS_OUTPUT.PUT_LINE('---');
152     DBMS_OUTPUT.PUT_LINE('Salariul maxim: '||DETALII_DEPT(J).SALARIU_MAX);
153   --afisam toti angajatii (există obligatoriu minim 1)
154   FOR K IN DETALII_DEPT(J).DETALII_ANGAJATI.FIRST..DETALII_DEPT(J).DETALII_ANGAJATI.LAST LOOP
155     DBMS_OUTPUT.PUT_LINE(DETALII_DEPT(J).DETALII_ANGAJATI(K).NUME
156     || ' ' || DETALII_DEPT(J).DETALII_ANGAJATI(K).SALARIU);
157   END LOOP;
158 END LOOP;
159 ELSE
160   DBMS_OUTPUT.PUT_LINE('Casa de discuri nu are departamente sau angajati');
161 END IF;
162 DBMS_OUTPUT.PUT_LINE('-----');
163 DBMS_OUTPUT.NEW_LINE;
164 EXCEPTION
165   WHEN NO_CD THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista casa de discuri cu acel id');
166   WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20001,'nu au fost gasite date');
167   WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20001,'au fost returnate prea multe linii');
168   WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20001,'alta eroare');
169 END ABOUT_CASA_DISCURI;
170
171 --dupa nume
172 PROCEDURE ABOUT_CASA_DISCURI (CD_NUME IN CASA_DISCURI.NUME%TYPE) IS
173   CD_ID CASA_DISCURI.ID_CASA_DISCURI%TYPE;
174   VCOUNT NUMBER(1);
175   NO_CD EXCEPTION;
176   DBMS_OUTPUT;
177
178 BEGIN
179   VCOUNT := 0;
180   NO_CD := FALSE;
181   FOR I IN 1..VCOUNT LOOP
182     SELECT COUNT(ID_CASA_DISCURI)
183     INTO VCOUNT
184     FROM CASA_DISCURI
185     WHERE UPPER(CASA_DISCURI.NUME) = UPPER(CD_NUME);
186
187   --ORIGATORIU EXISTA MAXIM 1 CD CU ACEL NUME
188   IF VCOUNT > 0 THEN
189     SELECT ID_CASA_DISCURI
190     INTO CD_ID
191     FROM CASA_DISCURI
192     WHERE UPPER(CASA_DISCURI.NUME) = UPPER(CD_NUME);
193     ABOUT_CASA_DISCURI(CD_ID);
194   ELSE
195     RAISE NO_CD;
196   END IF;
197
198 EXCEPTION
199   WHEN NO_CD THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista casa de discuri cu acel nume');
200 END ABOUT_CASA_DISCURI;
201
202 --ABOUT ALBUM
203 --NU SE PUNEREA TRUPEI CARE A LANSAZAT, TIPUL SI DATA LANSAZII DACA EXISTA
204 --SE STIU CE TRUPE AU CONTRIBUIT FOLOSIND FUNCTIA = TABEL INDEXAT, INDEX = ID_TRUPE
205 --LUNGIME TOTALA
206 --TOATE CANTECELE = TABEL INDEXAT
207
208 --dupa id
209 PROCEDURE ABOUT_ALBUM (VALBUM_ID IN ALBUM.ID_ALBUM%TYPE) IS
210   --declaratii
211   NUME_ALBUM ALBUM.NUME%TYPE;
212   NUME_TRUPE TRUPE.NUME%TYPE;
213   TIP_ALBUM ALBUM.TIP%TYPE;
214   DATA_ALBUM ALBUM.DATA_LANSAZIE%TYPE;
215   DURATA_ALBUM CANTEC.LUNGIME%TYPE;
216
217   LISTA_ID_TRUPE T_ID_TRUPE.CANTEC;
218   TEMP_NUME_TRUPE TRUPE.NUME%TYPE;
219
```

Script Output    Query Result

Task completed in 0.06 seconds

ERROR at line 1:

ORA-20001: Nu exista casa de discuri cu acel nume

ORA-06512: at "LUCA.EX13", line 166

ORA-06512: at line 2

Worksheet    Query Builder

```
174 VCOUNT NUMBER(1);
175 NO_CD EXCEPTION;
176 BEGIN
177   --DACA EXISTA
178   SELECT COUNT(ID_CASA_DISCURI)
179   INTO VCOUNT
180   FROM CASA_DISCURI
181   WHERE UPPER(CASA_DISCURI.NUME) = UPPER(CD_NUME);
182
183   --ORIGATORIU EXISTA MAXIM 1 CD CU ACEL NUME
184   IF VCOUNT > 0 THEN
185     SELECT ID_CASA_DISCURI
186     INTO CD_ID
187     FROM CASA_DISCURI
188     WHERE UPPER(CASA_DISCURI.NUME) = UPPER(CD_NUME);
189     ABOUT_CASA_DISCURI(CD_ID);
190   ELSE
191     RAISE NO_CD;
192   END IF;
193
194 EXCEPTION
195   WHEN NO_CD THEN RAISE_APPLICATION_ERROR(-20001,'Nu exista casa de discuri cu acel nume');
196 END ABOUT_CASA_DISCURI;
197
198 --ABOUT ALBUM
199 --NU SE PUNEREA TRUPEI CARE A LANSAZAT, TIPUL SI DATA LANSAZII DACA EXISTA
200 --SE STIU CE TRUPE AU CONTRIBUIT FOLOSIND FUNCTIA = TABEL INDEXAT, INDEX = ID_TRUPE
201 --LUNGIME TOTALA
202 --TOATE CANTECELE = TABEL INDEXAT
203
204 --dupa id
205 PROCEDURE ABOUT_ALBUM (VALBUM_ID IN ALBUM.ID_ALBUM%TYPE) IS
206   --declaratii
207   NUME_ALBUM ALBUM.NUME%TYPE;
208   NUME_TRUPE TRUPE.NUME%TYPE;
209   TIP_ALBUM ALBUM.TIP%TYPE;
210   DATA_ALBUM ALBUM.DATA_LANSAZIE%TYPE;
211   DURATA_ALBUM CANTEC.LUNGIME%TYPE;
212
213   LISTA_ID_TRUPE T_ID_TRUPE.CANTEC;
214   TEMP_NUME_TRUPE TRUPE.NUME%TYPE;
215
```

Recorduri

recorduri	Digital Nightmare	Tip: Album	Lansat de: TWRP	Featuring: Montaigne Tom Cardy	Data lansarii: 22-APR-24	Durata: 27.12 min
1	1 - 2.14 - Digital Nightmare					
2	2 - 4.29 - Content 4 U					
3	3 - 4.48 - HTMLove					
4	4 - 4.14 - Online					
5	5 - 3.56 - Dance Floor Cold War					
6	6 - 4.00 - Firewall					
7	7 - 4.51 - A Human's Touch					

Script Output    Query Result

More Details : https://docs.oracle.com/error-help/db/ora-20001/ https://docs.oracle.com/error-help/db/ora-06512/

PL/SQL procedure successfully completed.

Serba Ioana-Raluca  
Seria 24 Grupa 241

Worksheet | Query Builder

```

213
214     LISTA_ID_TRUPE T.ID_TRUPE_CANTEC;
215     TEMP_NUME TRUPA.NUME%TYPE;
216
217     TYPE T_FEAT_TRUPE IS TABLE OF NUMBER INDEX BY TRUPA.NUME%TYPE;
218     FEAT_TRUPE T.FEAT_TRUPE; --NUMELE TRUPELOR CARE AU CONTRIBUIT, INDEXATE DUPA NUMELE TRUPEI
219     --CA SA FIE UNICE
220     INDEX TRUPA.NUME%TYPE;
221
222     TYPE DETALII_CANTEC IS RECORD(
223         NUME_CANTEC.NUME%TYPE,
224         LUNGIME_CANTEC.LONGTIME%TYPE);
225
226     TYPE T_CANTECE_ALBUM IS TABLE OF DETALII_CANTEC INDEX BY PLS_INTEGER;
227     CANTECE_ALBUM T_CANTECE_ALBUM;
228
229     VCOUNT NUMBER(2);
230     NO_ALBUM EXCEPTION;
231
232     CURSOR CANTEC(VID_ALBUM ALBUM.ID_ALBUM%TYPE) IS
233         SELECT ID_CANTEC
234             FROM CANTEC_ALBUM
235             WHERE ID_ALBUM = VID_ALBUM;
236
237     BEGIN
238         --DACA EXISTA ALBUMUL
239         SELECT COUNT(ID_ALBUM)
240         INTO VCOUNT
241         FROM ALBUM
242         WHERE ID_ALBUM = VALBUM_ID;
243
244         IF VCOUNT = 0 THEN
245             RAISE NO_ALBUM;
246         END IF;
247
248         --CAUT NUME, SI RESTUL CESTIILOR DACA EXISTA
249         SELECT A.NUME, T.NUME, TIP, DATA_LANSARE
250             INTO NUME_ALBUM, NUME_TRUPA, TIP_ALBUM, DATA_ALBUM
251             FROM ALBUM A
252             JOIN TRUPA T ON A.ID_TRUPA = T.ID_TRUPA
253             WHERE ID_ALBUM = VALBUM_ID;
254

```

The Normal Album  
Tip: EP  
Lansat de: Will Wood  
Data lansarii: 10-JUN-20  
Durata: 13:43 min  
-----  
1 - 6.16 - Suburbia Overture  
2 - 4.01 - Laplace's Angel  
3 - 3.26 - Second-night-Zeier

Red Flags  
Tip: Single  
Lansat de: Tom Cardy  
Data lansarii: 12-MAR-23  
Durata: 4:12 min  
-----  
1 - 4.12 - Red Flags

Heartbreak in Stereo  
Tip: Album  
Lansat de: Frank Iero  
Data lansarii: 26-NOV-01  
Durata: 0 min  
-----  
Albumul nu are cantece inregistrate

Script Output | Query Result | Task completed in 0.184 seconds

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Worksheet | Query Builder

```

254
255     --afiam numele trupelor care au contribuit la cantece de pe album, daca exista
256     --trec prin toate cantecele, si folosind functia retin in tabel numele trupelor
257
258     FOR VCANTEC IN CANTECE(VALBUM_ID) LOOP
259         LISTA_ID_TRUPE := TRUPE_CANTEC(VCANTEC.ID_CANTEC);
260         IF LISTA_ID_TRUPE.COUNT > 0 THEN --in cazul in care albumul nu are cantece inregistrate
261             FOR I IN LISTA_ID_TRUPE.FIRST..LISTA_ID_TRUPE.LAST LOOP
262                 SELECT NUME
263                 INTO TEMP_NUME
264                 FROM TRUPA
265                 WHERE ID_TRUPA = LISTA_ID_TRUPE(I);
266                 IF TEMP_NUME != NUME_TRUPA THEN
267                     FEAT_TRUPE(TEMP_NUME) := 1; --PRACTIC, IMI PASA DOAR DE INDEX
268                 END IF;
269             END LOOP;
270         END IF;
271     END LOOP;
272
273
274     --si populez si tabelul de cantece
275     SELECT C.NUME, C.LUNGIME
276     BULK COLLECT INTO CANTECE_ALBUM
277     FROM CANTEC C
278     JOIN CANTEC_ALBUM CA ON C.ID_CANTEC = CA.ID_CANTEC
279     WHERE ID_ALBUM = VALBUM_ID;
280
281
282     --afiam durata totala
283     SELECT SUM(LUNGIME)
284     INTO DURATA_ALBUM
285     FROM CANTEC C
286     JOIN CANTEC_ALBUM CA ON CA.ID_CANTEC = C.ID_CANTEC
287     WHERE CA.ID_ALBUM = VALBUM_ID;
288
289     --AFISAM detaliiile albumului
290     DBMS_OUTPUT.PUT_LINE(NUME_ALBUM);
291     IF TIP_ALBUM IS NOT NULL THEN
292         DBMS_OUTPUT.PUT_LINE('Tip: ' || TIP_ALBUM);
293     END IF;
294     DBMS_OUTPUT.PUT_LINE('Lansat de: ' || NUME_TRUPA);
295

```

Script Output | Query Result | Task completed in 0.068 seconds

ERROR at line 1:  
ORA-20002: Nu exista albumul cu acel id  
ORA-06512: at "LUCA.EX13", line 294  
ORA-06512: at line 2

<https://docs.oracle.com/error-help/db/ora-20002/>

Serba Ioana-Raluca  
Seria 24 Grupa 241

```

295
296    --FEAT TRUPELE CARE AU CONTRIBUIT
297    IF FEAT_TRUPE.COUNT > 0 THEN
298        INDEX := FEAT_TRUPE.FIRST;
299        DBMS_OUTPUT.PUT_LINE('Featuring: '|| INDEX);
300        INDEX := FEAT_TRUPE.NEXT(INDEX);
301        WHILE INDEX IS NOT NULL LOOP
302            DBMS_OUTPUT.PUT_LINE('           '|| INDEX);
303            INDEX := FEAT_TRUPE.NEXT(INDEX);
304        END LOOP;
305    END IF;
306    IF DATA_ALBUM IS NOT NULL THEN
307        DBMS_OUTPUT.PUT_LINE('Data lansarii: '||DATA_ALBUM);
308    END IF;
309    DBMS_OUTPUT.PUT_LINE('Durata: '||NVL(DURATA_ALBUM,0)||' min');
310    DBMS_OUTPUT.PUT_LINE('-----');
311
312    IF CANTECE_ALBUM.COUNT > 0 THEN
313        FOR I IN CANTECE_ALBUM.FIRST..CANTECE_ALBUM.LAST LOOP
314            DBMS_OUTPUT.PUT_LINE(I||'-'||TO_CHAR(CANTECE_ALBUM(I).LUNGIME, 'FM999.00')||' - ';
315            || CANTECE_ALBUM(I).NUME); --+ FORMATARE sa fie mereu 2 zecimale
316        END LOOP;
317    ELSE
318        DBMS_OUTPUT.PUT_LINE('Albumul nu are cantece inregistrate');
319    END IF;
320    DBMS_OUTPUT.NEW_LINE;
321
322 EXCEPTION
323     WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Nu exista albumul cu acel id');
324     WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20002,'nu au fost gasite date');
325     WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20002,'au fost returnate prea multe linii');
326     WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20002,'alta eroare');
327 END ABOUT_ALBUM;
328
329
330 -- DUPA NUME
331 PROCEDURE ABOUT_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE) IS
332     VCOUNT NUMBER(2);
333     ALBUM_ID ALBUM.ID_ALBUM%TYPE;
334     TOO_MANY_ALBUM EXCEPTION;
335     NO_ALBUM EXCEPTION;
336     BEGIN
337
338     NO_ALBUM EXCEPTION;
339     BEGIN
340         SELECT COUNT(ID_ALBUM)
341             INTO VCOUNT
342             FROM ALBUM
343             WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);
344
345         IF VCOUNT > 1 THEN
346             RAISE TOO_MANY_ALBUM;
347         END IF;
348
349         IF VCOUNT = 1 THEN
350             SELECT ID_ALBUM
351                 INTO ALBUM_ID
352                 FROM ALBUM
353                 WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);
354             ABOUT_ALBUM(ALBUM_ID);
355         ELSE
356             RAISE NO_ALBUM;
357         END IF;
358     EXCEPTION
359         WHEN TOO_MANY_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Există mai multe albume cu acel nume. Folositi GA');
360         WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Nu există albume cu acel nume');
361     END ABOUT_ALBUM;
362
363     --ABOUT CANTEC
364     --NAME, LUNGIME
365     --NUMELE MUZICIENILOR (NU TRUPE!) CARE AU CONTRIBUIT LA CANTEC
366     -- + POZITIA IN TRUPA ) TABEL(ID_TRUPA)CU RECORD(NUME,POZITIE) -FOLOSIND FUNCTIA
367
368     --DUPA ID
369     PROCEDURE ABOUT_CANTEC (VCANTEC_ID IN CANTEC.ID_CANTEC%TYPE) IS
370         NUME_CANTEC CANTEC.NUME%TYPE;
371         LUNGIME_CANTEC CANTEC.LUNGIME%TYPE;
372         VCOUNT NUMBER(2);
373         NO_CANTEC EXCEPTION;
374
375         TYPE R_DETALII_MUZICIENI IS RECORD(
376             NUME MUZICIAN.NUME%TYPE,
377             POZITIE MEMBRU_TRUPA.PEZITIE%TYPE);
378         TYPE T_DETALII_MUZICIENI IS TABLE OF R_DETALII_MUZICIENI INDEX BY PLS_INTEGER;
379
380         R_DETALII_MUZICIENI RDETALII_MUZICIENI;
381         TDETALII_MUZICIENI TDETALII_MUZICIENI;
382
383         CURSOR CANTEC_CURSOR IS
384             SELECT * FROM CANTEC
385             WHERE ID_CANTEC = VCANTEC_ID;
386
387         CANTEC_CURSOR%ROWTYPE CANTEC_ROW;
388
389         FOR CANTEC_CURSOR IN CANTEC_CURSOR LOOP
390             CANTEC_ROW := CANTEC_CURSOR;
391             EXIT;
392         END LOOP;
393
394         FOR I IN 1..VCOUNT LOOP
395             R_DETALII_MUZICIENI.NUME := CANTEC_ROW.NUME;
396             R_DETALII_MUZICIENI.POZITIE := CANTEC_ROW.POZITIE;
397             TDETALII_MUZICIENI.EXTEND;
398             TDETALII_MUZICIENI(ICONT) := R_DETALII_MUZICIENI;
399         END LOOP;
400
401         DBMS_OUTPUT.PUT_LINE('-----');
402         DBMS_OUTPUT.PUT_LINE('Detalii muzicieni pentru cantec cu id: '||VCANTEC_ID);
403         DBMS_OUTPUT.PUT_LINE('-----');
404
405         FOR I IN 1..TDETALII_MUZICIENI.COUNT LOOP
406             DBMS_OUTPUT.PUT_LINE(TDETALII_MUZICIENI(ICONT).NUME||'-'||TDETALII_MUZICIENI(ICONT).POZITIE);
407         END LOOP;
408
409     EXCEPTION
410         WHEN NO_CANTEC THEN
411             RAISE_APPLICATION_ERROR(-20002,'Nu există canteci cu acel id');
412     END ABOUT_CANTEC;
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490

```

Script Output | Query Result | Task completed in 0.067 seconds

ERROR at line 1:  
ORA-20002: Nu există albume cu acel nume  
ORA-06512: at "LUCA.EX13", line 329  
ORA-06512: at line 2

<https://docs.oracle.com/error-help/db/ora-20002/>

```

335     NO_ALBUM EXCEPTION;
336     BEGIN
337         --DACA EXISTA
338         SELECT COUNT(ID_ALBUM)
339             INTO VCOUNT
340             FROM ALBUM
341             WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);
342
343         IF VCOUNT > 1 THEN
344             RAISE TOO_MANY_ALBUM;
345         END IF;
346
347         IF VCOUNT = 1 THEN
348             SELECT ID_ALBUM
349                 INTO ALBUM_ID
350                 FROM ALBUM
351                 WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);
352             ABOUT_ALBUM(ALBUM_ID);
353         ELSE
354             RAISE NO_ALBUM;
355         END IF;
356     EXCEPTION
357         WHEN TOO_MANY_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Există mai multe albume cu acel nume. Folositi GA');
358         WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20002,'Nu există albume cu acel nume');
359     END ABOUT_ALBUM;
360
361     --ABOUT CANTEC
362     --NAME, LUNGIME
363     --NUMELE MUZICIENILOR (NU TRUPE!) CARE AU CONTRIBUIT LA CANTEC
364     -- + POZITIA IN TRUPA ) TABEL(ID_TRUPA)CU RECORD(NUME,POZITIE) -FOLOSIND FUNCTIA
365
366     --DUPA ID
367     PROCEDURE ABOUT_CANTEC (VCANTEC_ID IN CANTEC.ID_CANTEC%TYPE) IS
368         NUME_CANTEC CANTEC.NUME%TYPE;
369         LUNGIME_CANTEC CANTEC.LUNGIME%TYPE;
370         VCOUNT NUMBER(2);
371         NO_CANTEC EXCEPTION;
372
373         TYPE R_DETALII_MUZICIENI IS RECORD(
374             NUME MUZICIAN.NUME%TYPE,
375             POZITIE MEMBRU_TRUPA.PEZITIE%TYPE);
376         TYPE T_DETALII_MUZICIENI IS TABLE OF R_DETALII_MUZICIENI INDEX BY PLS_INTEGER;
377
378         RDETALII_MUZICIENI RDETALII_MUZICIENI;
379         TDETALII_MUZICIENI TDETALII_MUZICIENI;
380
381         CURSOR CANTEC_CURSOR IS
382             SELECT * FROM CANTEC
383             WHERE ID_CANTEC = VCANTEC_ID;
384
385         CANTEC_CURSOR%ROWTYPE CANTEC_ROW;
386
387         FOR CANTEC_CURSOR IN CANTEC_CURSOR LOOP
388             CANTEC_ROW := CANTEC_CURSOR;
389             EXIT;
390         END LOOP;
391
392         FOR I IN 1..VCOUNT LOOP
393             RDETALII_MUZICIENI.NUME := CANTEC_ROW.NUME;
394             RDETALII_MUZICIENI.POZITIE := CANTEC_ROW.POZITIE;
395             TDETALII_MUZICIENI.EXTEND;
396             TDETALII_MUZICIENI(ICONT) := RDETALII_MUZICIENI;
397         END LOOP;
398
399         DBMS_OUTPUT.PUT_LINE('-----');
400         DBMS_OUTPUT.PUT_LINE('Detalii muzicieni pentru cantec cu id: '||VCANTEC_ID);
401         DBMS_OUTPUT.PUT_LINE('-----');
402
403         FOR I IN 1..TDETALII_MUZICIENI.COUNT LOOP
404             DBMS_OUTPUT.PUT_LINE(TDETALII_MUZICIENI(ICONT).NUME||'-'||TDETALII_MUZICIENI(ICONT).POZITIE);
405         END LOOP;
406
407     EXCEPTION
408         WHEN NO_CANTEC THEN
409             RAISE_APPLICATION_ERROR(-20002,'Nu există canteci cu acel id');
410     END ABOUT_CANTEC;
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490

```

Script Output | Query Result | Task completed in 0.052 seconds

ERROR at line 1:  
ORA-20002: Există mai multe albume cu acel nume. Folositi GASESTE\_ALBUM pentru a găsi id-ul albumului pe care îl doriti  
ORA-06512: at "LUCA.EX13", line 328  
ORA-06512: at line 2

<https://docs.oracle.com/error-help/db/ora-20002/>

Serba Ioana-Raluca  
Seria 24 Grupa 241

Worksheet | Query Builder

```

375      POSITIE MEMBRU_TRUPA.POSITIE%TYPE);
376      TYPE T_DETALII_MUZICIENI IS TABLE OF R_DETALII_MUZICIENI INDEX BY PLS_INTEGER;
377
378      TYPE R_TRUPA_MUZICIENI IS RECORD(
379          TRUPA_ID TRUPA.ID TRUPA%TYPE,
380          MUZICIENI T_DETALII_MUZICIENI); --pt fiecare id trupa, numele si pozitia muzicienilor
381      TYPE T_TRUPA_MUZICIENI IS TABLE OF R_TRUPA_MUZICIENI INDEX BY PLS_INTEGER;
382
383      MUZICIENI_CANTEC T_TRUPA_MUZICIENI;
384
385      -- FUNCTIE CARE RETURNEAZA MUZICIENII UNUI CANTEC SI FOLOSESTE TRUPE_CANTEC
386      FUNCTION F_MUZICIENI_CANTEC (VID_CANTEC CANTEC.ID_CANTEC%TYPE)
387          RETURN T_TRUPA_MUZICIENI IS
388          TRUPA_MUZICIENI;
389          ID_TRUPE_CANTEC T_ID_TRUPE_CANTEC;
390
391      BEGIN
392          ID_TRUPE_CANTEC := TRUPE_CANTEC(VID_CANTEC); --aici am idurile trupelor care au contribuit la acest c
393          IF ID_TRUPE_CANTEC.COUNT>0 THEN --in caz ca cantecul nu are trupe inregistrate
394              FOR I IN ID_TRUPE_CANTEC.FIRST..ID_TRUPE_CANTEC.LAST LOOP
395                  TRUPA_MUZICIENI(I).TRUPA_ID := ID_TRUPE_CANTEC(I);--PENTRU FIECARA ID DE TRUPA
396                  SELECT M.NUME, MT.POSITIE --IMI SUNT TOȚI MUZICIENII TRUPEI
397                  BULK COLLECT INTO TRUPA_MUZICIENI(I).MUZICIENI
398                  FROM MUZICIAN M
399                  JOIN MEMBRI_TRUPA MT ON M.ID_MUZICIAN = MT.ID_MUZICIAN
400                  WHERE MT.ID_TRUPA = ID_TRUPE_CANTEC(I);
401              END LOOP;
402          END IF;
403          RETURN TRUPA_MUZICIENI;
404      END F_MUZICIENI_CANTEC;
405
406      BEGIN
407          --DACA EXISTA CANTECTUL
408          SELECT COUNT(ID_CANTEC)
409          INTO VCOUNT
410          FROM CANTEC
411          WHERE ID_CANTEC = VCANTEC_ID;
412
413          IF VCOUNT = 0 THEN
414              RAISE NO_CANTEC;
415          END IF;
416
417          --CAUT NUME, SI RESTUL CHESTIILOR DACA EXISTA

```

Script Output | Query Result | Task completed in 0.049 seconds

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Messages - Log

```

```

414      END IF;
415
416      --CAUT NUME, SI RESTUL CHESTIILOR DACA EXISTA
417      SELECT NUME, LUNGIME
418      INTO NUME_CANTEC, LUNGIME_CANTEC
419      FROM CANTEC C
420      WHERE ID_CANTEC = VCANTEC_ID;
421
422      --POPULEZ TABELUL CU DETALIILE MUZICIENILOR
423      MUZICIENI_CANTEC := F_MUZICIENI_CANTEC(VCANTEC_ID);
424
425      --AFISEZ FRUMOS
426      DBMS_OUTPUT.PUT_LINE(NUME_CANTEC);
427      DBMS_OUTPUT.PUT_LINE('Durata: '||TO_CHAR(LUNGIME_CANTEC,'FM9999.00')|||' min');
428      DBMS_OUTPUT.PUT_LINE('-----');
429      DBMS_OUTPUT.PUT_LINE('Credits:');
430      IF MUZICIENI_CANTEC.COUNT > 0 THEN
431          FOR I IN MUZICIENI_CANTEC.FIRST..MUZICIENI_CANTEC.LAST LOOP
432              FOR J IN MUZICIENI_CANTEC(I).MUZICIENI.FIRST..MUZICIENI_CANTEC(I).MUZICIENI.LAST LOOP
433                  DBMS_OUTPUT.PUT_LINE(MUZICIENI_CANTEC(I).MUZICIENI(J).NUME ||| ' - '|||
434                  MUZICIENI_CANTEC(I).MUZICIENI(J).POZITIE );
435              END LOOP;
436          END LOOP;
437          ELSE DBMS_OUTPUT.PUT_LINE('Cantecul nu are muzicieni inregistrati');
438      END IF;
439
440
441      EXCEPTION
442          WHEN NO_CANTEC THEN RAISE_APPLICATION_ERROR(-20003,'Nu exista cantec cu acel id');
443          WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR (-20003,'nu au fost gasite date');
444          WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR (-20003,'au fost returnate prea multe linii');
445          --WHEN OTHERS THEN RAISE_APPLICATION_ERROR (-20003,'alta eroare');
446
447      END ABOUT_CANTEC;
448
449      --dupa nume
450      PROCEDURE ABOUT_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE) IS
451          VCOUNT NUMBER(2);
452          CANTEC_ID CANTEC.ID_CANTEC%TYPE;
453          TOO_MANY_CANTEC EXCEPTION;
454          NO_CANTEC EXCEPTION;
455          BEGIN
456              --DACA EXISTA

```

Script Output | Query Result | Task completed in 0.052 seconds

```

*
ERROR at line 1:
ORA-20003: Nu exista cantec cu acel id
ORA-06512: at "LUCA.EX13", line 413
ORA-06512: at line 2

https://docs.oracle.com/error-help/db/ora-20003/

```

Serba Ioana-Raluca  
Seria 24 Grupa 241

Worksheet    Query Builder

```

451:     CANTEC_ID_CANTEC.ID_CANTEC%TYPE;
452:     TOO_MANY_CANTEC EXCEPTION;
453:     NO_CANTEC EXCEPTION;
454: BEGIN
455:     BEGIN
456:         --DACA EXISTA
457:         SELECT COUNT(ID_CANTEC)
458:             INTO VCOUNT
459:             FROM CANTEC
460:             WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);
461:
462:         IF VCOUNT > 1 THEN
463:             RAISE TOO_MANY_CANTEC;
464:         END IF;
465:
466:         IF VCOUNT = 1 THEN
467:             SELECT ID_CANTEC
468:                 INTO CANTEC_ID
469:                 FROM CANTEC
470:                 WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);
471:             ABOUT_CANTEC(CANTEC_ID);
472:         ELSE
473:             RAISE NO_CANTEC;
474:         END IF;
475:     EXCEPTION
476:         WHEN TOO_MANY_CANTEC THEN RAISE_APPLICATION_ERROR(-20003,'Exista mai multe cantece cu acel nume. Folositi GASESTE_ALBUM pentru a gasi id-ul canticului pe care il doriti');
477:         WHEN NO_CANTEC THEN RAISE_APPLICATION_ERROR(-20003,'Nu exista cantece cu acel nume');
478:     END ABOUT_CANTEC;
479:
480: --procedurile de gasit id-uri la album si cantic
481: PROCEDURE GASESTE_ALBUM (VALBUM_NUME IN ALBUM.NUME%TYPE) IS
482:     NO_ALBUM EXCEPTION;
483:     VCOUNT NUMBER(2);
484:
485:     CURSOR ALBUM (VNUME IN ALBUM.NUME%TYPE) IS
486:         SELECT A.ID_ALBUM, A.NUME, T.NUME AS NUME_TRUPA, TIP
487:             FROM ALBUM A
488:             JOIN TRUPA T ON A.ID_TRUPA = T.ID_TRUPA
489:             WHERE UPPER(A.NUME) = UPPER(VNUME);
490:
491:     BEGIN
492:         SELECT COUNT(ID_ALBUM)
493:             INTO VCOUNT
494:             FROM ALBUM

```

Script Output x | Query Result x | Task completed in 0.054 seconds

\*  
ERROR at line 1:  
ORA-20003: Nu exista cantece cu acel nume  
ORA-06512: at "LUCA.EX13", line 448  
ORA-06512: at line 2

<https://docs.oracle.com/error-help/db/ora-20003/>

Worksheet    Query Builder

```

455:     WHERE UPPER(ALBUM.NUME) = UPPER(VALBUM_NUME);
456:
457:     IF VCOUNT = 0 THEN
458:         RAISE NO_ALBUM;
459:     END IF;
460:
461:     FOR ALBM IN ALBUM (VALBUM_NUME) LOOP
462:         DBMS_OUTPUT.PUT_LINE('ID: '|||ALBM.ID_ALBUM);
463:         DBMS_OUTPUT.PUT_LINE('Nume: '|||ALBM.NUME);
464:         DBMS_OUTPUT.PUT_LINE('Nume trupa: '|||ALBM.NUME_TRUPA);
465:         DBMS_OUTPUT.PUT_LINE('Tip: '|||ALBM.TIP);
466:         DBMS_OUTPUT.NEW_LINE;
467:     END LOOP;
468: EXCEPTION
469:     WHEN NO_ALBUM THEN RAISE_APPLICATION_ERROR(-20022,'Nu exista albume cu acel nume');
470: END GASESTE_ALBUM;
471:
472: PROCEDURE GASESTE_CANTEC (VCANTEC_NUME IN CANTEC.NUME%TYPE) IS
473:     NO_CANTEC EXCEPTION;
474:     VCOUNT NUMBER(2);
475:
476:     CURSOR CANTECE (VNUME IN CANTEC.NUME%TYPE) IS
477:         SELECT ID_CANTEC, NUME, LUNGIME
478:             FROM CANTEC C
479:             WHERE UPPER(C.NUME) = UPPER(VNUME);
480:
481:     CURSOR NUME_TRUPE (VID_CANTEC CANTEC.ID_CANTEC%TYPE) IS
482:         SELECT NUME
483:             FROM TRUPA T
484:             JOIN TRUPA_CANTEC TC ON TC.ID_TRUPA = T.ID_TRUPA
485:             WHERE TC.ID_CANTEC = VID_CANTEC;
486:
487:     BEGIN
488:         SELECT COUNT(ID_CANTEC)
489:             INTO VCOUNT
490:             FROM CANTEC
491:             WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);
492:
493:         IF VCOUNT = 0 THEN
494:             RAISE NO_CANTEC;
495:         END IF;

```

Script Output x | Query Result x | Task completed in 0.055 seconds

\*  
ERROR at line 1:  
ORA-20003: Exista mai multe cantece cu acel nume. Folositi GASESTE\_CANTEC pentru a gasi id-ul cantecului pe care il doriti  
ORA-06512: at "LUCA.EX13", line 447  
ORA-06512: at line 2

<https://docs.oracle.com/error-help/db/ora-20003/>

Serba Ioana-Raluca  
Seria 24 Grupa 241

```

519  WHERE UPPER(C.NUME) = UPPER(VNUME);
520
521  CURSOR NUME_TRUPE (VID_CANTEC CANTEC.ID_CANTEC%TYPE) IS
522    SELECT NUME
523    FROM TRUPA T
524    JOIN TRUPA_CANTEC TC ON TC.ID_TRUPA = T.ID_TRUPA
525    WHERE TC.ID_CANTEC = VID_CANTEC;
526
527  BEGIN
528    SELECT COUNT(ID_CANTEC)
529    INTO VCOUNT
530    FROM CANTEC
531    WHERE UPPER(CANTEC.NUME) = UPPER(VCANTEC_NUME);
532
533    IF VCOUNT = 0 THEN
534      RAISE NO_CANTEC;
535    END IF;
536
537    FOR CT IN CANTEC(VCANTEC_NUME) LOOP
538      DBMS_OUTPUT.PUT_LINE('ID: '||CT.ID_CANTEC);
539      DBMS_OUTPUT.PUT_LINE('Name: '||CT.NUME);
540      DBMS_OUTPUT.PUT_LINE('Lungime: '||TO_CHAR(CT.LUNGIME,'FM9999.00')||' min');
541      DBMS_OUTPUT.PUT_LINE('Trupe: '||CT.ID_TRUPA);
542      FOR CTT IN NUME_TRUPE(CT.ID_CANTEC) LOOP
543        DBMS_OUTPUT.PUT_LINE(CTT.NUME);
544      END LOOP;
545      DBMS_OUTPUT.NEW_LINE;
546    END LOOP;
547  EXCEPTION
548    WHEN NO_CANTEC THEN RAISE_APPLICATION_ERROR(-20033,'Nu exista cantece cu acel nume');
549  END GASESTE_CANTEC;
550
551 END EX13;
552 /
553 BEGIN
554   EX13.GASESTE_CANTEC('The No Pants Dance');
555 END;
556 /
557 select * from cantec;
558
559
560

```

PL/SQL procedure successfully completed.

```

553 BEGIN
554   EX13.GASESTE_CANTEC('woowow');
555 END;
556 /
557 select * from cantec;
558
559
560

```

Script Output x | Query Result x

Task completed in 0.05 seconds

\*

ERROR at line 1:  
ORA-20033: Nu exista cantece cu acel nume  
ORA-06512: at "LUCA.EX13", line 519  
ORA-06512: at line 2

```

553 BEGIN
554   EX13.GASESTE_ALBUM('woowow');
555 END;
556 /
557 select * from cantec;
558
559
560

```

Script Output x | Query Result x

Task completed in 0.05 seconds

\*

ERROR at line 1:  
ORA-20022: Nu exista albume cu acel nume  
ORA-06512: at "LUCA.EX13", line 480  
ORA-06512: at line 2