1. Cerinta:

Sa se formuleze in limbaj natural 4 probleme (si sa se rezolve) adaptate modelului de date personal, in care sa se foloseasca cursoare clasice, ciclu cursoare, ciclu cursoare cu subcereri si expresii cursor.

2. Crearea tabelelor si inserarea datelor:

```
CREATE TABLE MANAGER (
    id_manager NUMBER(4) PRIMARY KEY,
    nume VARCHAR(30) NOT NULL,
    nr_telefon VARCHAR(16) NOT NULL UNIQUE,
    mail VARCHAR(40) NOT NULL UNIQUE,
    adresa VARCHAR(80) NOT NULL
);
INSERT INTO MANAGER
VALUES(1, 'Quinn Lester', '+12138290401', 'lesterr.quinn3@gmail.com', '5759 Saturn St, Los Angeles, California(CA), 90019');
INSERT INTO MANAGER
VALUES(2, 'Roy Phillips', '+12137279163', 'roy_phillips173@gmail.com', '6736 S Sherbourne Dr, Los Angeles, California(CA), 90056');
INSERT INTO MANAGER
VALUES(3, 'Malcolm Hurst', '+15109128437', 'malcolm.hurstt0504@outlook.com', '352 Garretson Ave, Rodeo, California(CA), 94572');
INSERT INTO MANAGER
VALUES(4, 'Edward Austin', '+12069180936', 'austinnedward_13@gmail.com', '13021 25th Ave NE, Seattle, Washington(WA), 98125');
INSERT INTO MANAGER
VALUES(5, 'Anderson Rose', '+16172918745', 'anderson_rose1969@outlook.com', '786 Blue Hill Ave Boston, Massachusetts(MA), 02124');
INSERT INTO MANAGER
```

```
VALUES(6, 'Kevin Henson', '+13363351992', 'kevin.henson15@gmail.com', '748 Havanna Street, 27401, Greensboro, North Carolina, US');


CREATE TABLE TRUPA (
   id_trupa NUMBER(4) PRIMARY KEY,
   id_manager NUMBER(4) NOT NULL,
   nume VARCHAR(30) NOT NULL UNIQUE,
   data_infiintare DATE NOT NULL,
   FOREIGN KEY (id_manager) REFERENCES MANAGER(id_manager)
);
INSERT INTO TRUPA
VALUES(1, 1, 'Red Hot Chili Peppers', '15-NOV-1982');
INSERT INTO TRUPA
VALUES(2, 2, 'Metallica', '11-MAY-1981');
INSERT INTO TRUPA
VALUES(3, 1, 'Green Day', '26-JUN-1986');
INSERT INTO TRUPA
VALUES(4, 4, 'Heart', '2-FEB-1973');
INSERT INTO TRUPA
VALUES(5, 5, 'Aerosmith', '9-APR-1970');


CREATE TABLE ALBUM (
   id_album NUMBER(4) PRIMARY KEY,
   id_trupa NUMBER(4) NOT NULL,
   titlu VARCHAR(30) NOT NULL,
   data_lansare DATE NOT NULL,
   nr_piese NUMBER(2) NOT NULL,
    FOREIGN KEY (id_trupa) REFERENCES TRUPA(id_trupa)
);
INSERT INTO ALBUM
VALUES(10, 1, 'Californication', '08-JUN-1999', 15);
INSERT INTO ALBUM
VALUES(11, 1, 'By The Way', '09-JUL-2002', 16);
INSERT INTO ALBUM
VALUES(12, 1, 'Stadium Arcadium', '09-MAY-2006', 28);
INSERT INTO ALBUM
VALUES(13, 2, 'Master of Puppets', '03-MAR-1986', 10);
INSERT INTO ALBUM
VALUES(14, 2, 'Metallica', '12-AUG-1991', 12);
INSERT INTO ALBUM
VALUES(15, 3, 'American Idiot', '21-SEP-2004', 13);
INSERT INTO ALBUM
VALUES(16, 3, 'Dookie', '01-FEB-1994', 15);
INSERT INTO ALBUM
VALUES(17, 4, 'Dreamboat Annie', '14-FEB-1976', 10);
INSERT INTO ALBUM
VALUES(18, 4, 'Bad Animals', '15-MAY-1987', 10);
INSERT INTO ALBUM
VALUES(19, 5, 'Get A Grip', '20-APR-1993', 14);
INSERT INTO ALBUM
VALUES(20, 5, 'Toys in the Attic', '08-APR-1975', 9);


CREATE TABLE PIESA (
    id_piesa NUMBER(4) PRIMARY KEY,
    id_album NUMBER(4) NOT NULL,
    titlu VARCHAR(30) NOT NULL,
    durata NUMBER(4) NOT NULL,
    FOREIGN KEY (id_album) REFERENCES ALBUM(id_album)
);
INSERT INTO PIESA
VALUES(1, 10, 'Road Trippin', 205);
INSERT INTO PIESA
VALUES(2, 10, 'Scar Tissue', 216);
INSERT INTO PIESA
VALUES(3, 11, 'Tear', 317);
INSERT INTO PIESA
VALUES(4, 11, 'By the Way', 338);
INSERT INTO PIESA
VALUES(5, 12, 'Snow (Hey Oh)', 335);
INSERT INTO PIESA
VALUES(6, 12, 'Dani California', 283);
INSERT INTO PIESA
VALUES(7, 13, 'Master Of Puppets', 515);
INSERT INTO PIESA
VALUES(8, 13, 'Orion', 502);
INSERT INTO PIESA
VALUES(9, 14, 'Nothing Else Matters', 389);
INSERT INTO PIESA
VALUES(10, 14, 'Enter Sandman', 332);
INSERT INTO PIESA
VALUES(11, 15, 'Boulevard of Broken Dreams', 494);
INSERT INTO PIESA
```

```
VALUES(12, 15, 'Wake Me Up When September Ends', 286);
INSERT INTO PIESA
VALUES(13, 16, 'Basket Case', 183);
INSERT INTO PIESA
VALUES(14, 16, 'When I Come Around', 178);
INSERT INTO PIESA
VALUES(15, 17, 'Crazy On You', 293);
INSERT INTO PIESA
VALUES(16, 17, 'Magic Man', 328);
INSERT INTO PIESA
VALUES(17, 18, 'Alone', 219);
INSERT INTO PIESA
VALUES(18, 18, 'Who Will You Run To', 246);
INSERT INTO PIESA
VALUES(19, 19, 'Cryin', 309);
INSERT INTO PIESA
VALUES(20, 19, 'Crazy', 317);
INSERT INTO PIESA
VALUES(21, 20, 'Walk This Way', 221);
INSERT INTO PIESA
VALUES(22, 20, 'Sweet Emotion', 275);


CREATE TABLE SPONSOR (
   id_sponsor NUMBER(4) PRIMARY KEY,
   nume VARCHAR(30) NOT NULL,
   nr_telefon VARCHAR(16) NOT NULL UNIQUE,
   mail VARCHAR(40) NOT NULL UNIQUE
);
INSERT INTO SPONSOR
VALUES(1, 'Alia Klein', '+19257438187', 'aliaa_klein50@gmail.com');
INSERT INTO SPONSOR
VALUES(2, 'Julius Wallace', '+14258219807', 'wallace.julius94@outlook.com');
INSERT INTO SPONSOR
VALUES(3, 'Della Maxwell', '+14233693822', 'dellamaxwell_1993@gmail.com');
INSERT INTO SPONSOR
VALUES(4, 'Angel Wilson', '+15702260731', 'angellwilson9214@gmail.com');
INSERT INTO SPONSOR
VALUES(5, 'Isabelle Huang', '+16014286195', 'isabellehuang.32@outlook.com');
INSERT INTO SPONSOR
VALUES(6, 'Lewis Clayton', '+15059849972', 'clayton.lewis909@gmail.com');
INSERT INTO SPONSOR
VALUES(7, 'Kyson McDowell', '+18166303886', 'kyson_mcdowell44@outlook.com');


CREATE TABLE CONCERT (
   id_concert NUMBER(4) PRIMARY KEY,
   data DATE NOT NULL
);
INSERT INTO CONCERT
VALUES(100, '05-AUG-2024');
INSERT INTO CONCERT
VALUES(110, '28-DEC-2024');
INSERT INTO CONCERT
VALUES(120, '12-OCT-2024');
INSERT INTO CONCERT
```

```
VALUES(130, '02-NOV-2024');
INSERT INTO CONCERT
VALUES(140, '14-FEB-2025');


CREATE TABLE SPONSORIZARE (
    id_concert NUMBER(4) NOT NULL,
    id_sponsor NUMBER(4) NOT NULL,
    PRIMARY KEY (id_concert, id_sponsor),
    FOREIGN KEY (id_concert) REFERENCES CONCERT(id_concert),
    FOREIGN KEY (id_sponsor) REFERENCES SPONSOR(id_sponsor)
);
INSERT INTO SPONSORIZARE
VALUES(100, 1);
INSERT INTO SPONSORIZARE
VALUES(130, 1);
INSERT INTO SPONSORIZARE
VALUES(100, 4);
INSERT INTO SPONSORIZARE
VALUES(140, 7);
INSERT INTO SPONSORIZARE
VALUES(110, 5);
INSERT INTO SPONSORIZARE
VALUES(120, 5);
INSERT INTO SPONSORIZARE
VALUES(100, 5);
INSERT INTO SPONSORIZARE
VALUES(110, 6);
```

```sql
INSERT INTO SPONSORIZARE
VALUES(100, 2);
INSERT INTO SPONSORIZARE
VALUES(140, 2);



CREATE TABLE CLIENT (
    id_client NUMBER(4) PRIMARY KEY,
    nume VARCHAR(30) NOT NULL,
    nr_telefon VARCHAR(16) NOT NULL UNIQUE,
    mail VARCHAR(40) NOT NULL UNIQUE,
    adresa VARCHAR(80) NOT NULL
);
INSERT INTO CLIENT
VALUES(22, 'Bellamy Knight', '+17167984922', 'bellamy.knight.480@gmail.com', '400 Commercial St, Medina, New York, 14103');
INSERT INTO CLIENT
VALUES(23, 'Leighton Huffman', '+490821472510', 'leighton_huffman1329@outlook.com', 'Mellingburgredder 84, Happurg, Freistaat Bayern, Germany, 91230');
INSERT INTO CLIENT
VALUES(24, 'Nancy Schultz', '+610246004601 ', 'nancyschultz319@gmail.com', '161 Keilor Rd, Essendon VIC 3040, Melbourne, Australia');
INSERT INTO CLIENT
VALUES(25, 'Eden Guzman', '+330174277258', 'eden.guzman22@outlook.com', '76 Rue des Belles Feuilles, 75116, Paris, France');
INSERT INTO CLIENT
VALUES(26, 'Leonard Bates', '+81997395728', 'leonard.bates09@gmail.com', 'Senkawa-dori Avenue, 176-0006, Tokyo, Japan');
```

```
CREATE TABLE CONTRACT (
   id_client NUMBER(4) NOT NULL,
   id_trupa NUMBER(4) NOT NULL,
   id_concert NUMBER(4) NOT NULL,
   data_semnare DATE NOT NULL,
   onorariu NUMBER(5) DEFAULT 5000,
   PRIMARY KEY (id_client, id_trupa, id_concert),
   FOREIGN KEY (id_client) REFERENCES CLIENT(id_client),
   FOREIGN KEY (id_trupa) REFERENCES TRUPA(id_trupa),
   FOREIGN KEY (id_concert) REFERENCES CONCERT(id_concert)
);
INSERT INTO CONTRACT
VALUES(22, 1, 100, '01-JUN-2024', 12500);
INSERT INTO CONTRACT (id_client, id_trupa, id_concert, data_semnare)
VALUES(22, 2, 100, '14-JUL-2024');
INSERT INTO CONTRACT
VALUES(26, 4, 140, '01-SEP-2024', 15000);
INSERT INTO CONTRACT
VALUES(23, 3, 110, '08-DEC-2024', 14000);
INSERT INTO CONTRACT
VALUES(25, 1, 130, '25-OCT-2024', 10000);
INSERT INTO CONTRACT
VALUES(26, 5, 140, '25-OCT-2024', 17500);
INSERT INTO CONTRACT (id_client, id_trupa, id_concert, data_semnare)
VALUES(23, 5, 120, '10-SEP-2024');
INSERT INTO CONTRACT
VALUES(23, 5, 110, '03-AUG-2024', 9000);
INSERT INTO CONTRACT
```

```
VALUES(26, 2, 140, '07-DEC-2024', 13500);
INSERT INTO CONTRACT
VALUES(25, 3, 130, '19-AUG-2024', 13500);
COMMIT;
```

3. 4 probleme folosind cursoare (rezolvare + explicatie + poze):

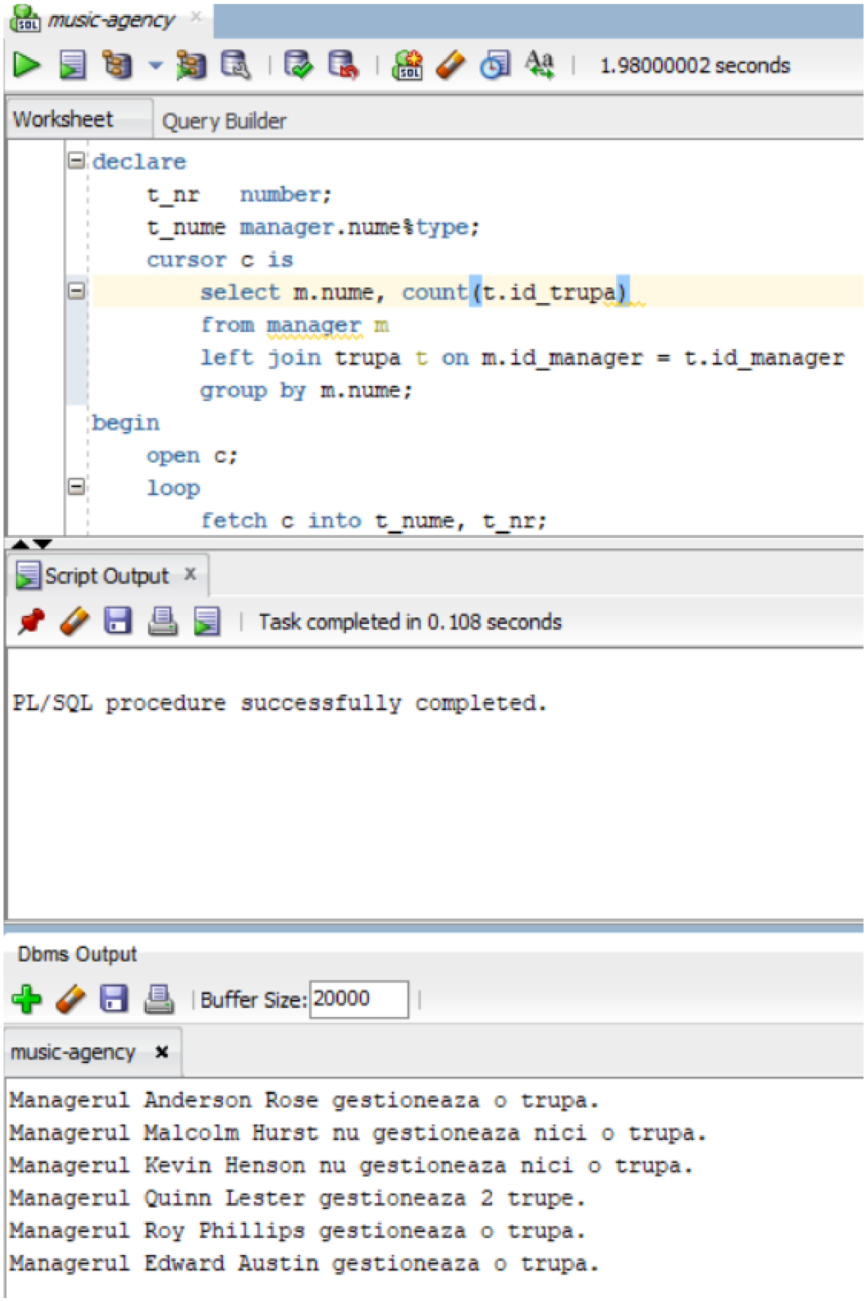
CURSOR SIMPLU: in modelul de date personal, fiecare TRUPA are un singur MANAGER, iar un MANAGER poate gestiona 0 sau mai multe TRUPE.

Am declarat un cursor pentru a retine numele si numarul de trupe gestionate pentru fiecare manager. Am deschis cursorul cu OPEN, am folosit un LOOP (iesirea din LOOP se face cand nu mai exista date catre care sa pointeze cursorul), am preluat datele din fiecare linie folosind FETCH INTO in nste variabile auxiliare, am inchis cursorul cu CLOSE.

In functie de numarul de trupe gestionate, am tratat 3 cazuri folosind un IF: daca managerul gestioneaza 0 trupe se va afisa un mesaj corespunzator; la fel si pentru 1 sau mai multe trupe.

```
declare
   t_nr   number;
   t_nume manager.nume%type;
   cursor c is
      select m.nume, count(t.id_trupa)
      from manager m
      left join trupa t on m.id_manager = t.id_manager
      group by m.nume;
begin
   open c;
   loop
      fetch c into t_nume, t_nr;
      exit when c%notfound;
      if t_nr = 0 then
         dbms_output.put_line('Managerul ' || t_nume || ' nu gestioneaza nici o trupa.');
      elsif t_nr = 1 then
         dbms_output.put_line('Managerul ' || t_nume || ' gestioneaza o trupa.');
```

```
        else
            dbms_output.put_line('Managerul ' || t_nume || ' gestioneaza ' || t_nr || ' trupe.');
        end if;
    end loop;
    close c;
end;
```



```
declare
    t_nr    number;
    t_nume manager.nume%type;
    cursor c is
        select m.nume, count(t.id_trupa)
        from manager m
        left join trupa t on m.id_manager = t.id_manager
        group by m.nume;
begin
    open c;
    loop
        fetch c into t_nume, t_nr;
```

```
PL/SQL procedure successfully completed.
```

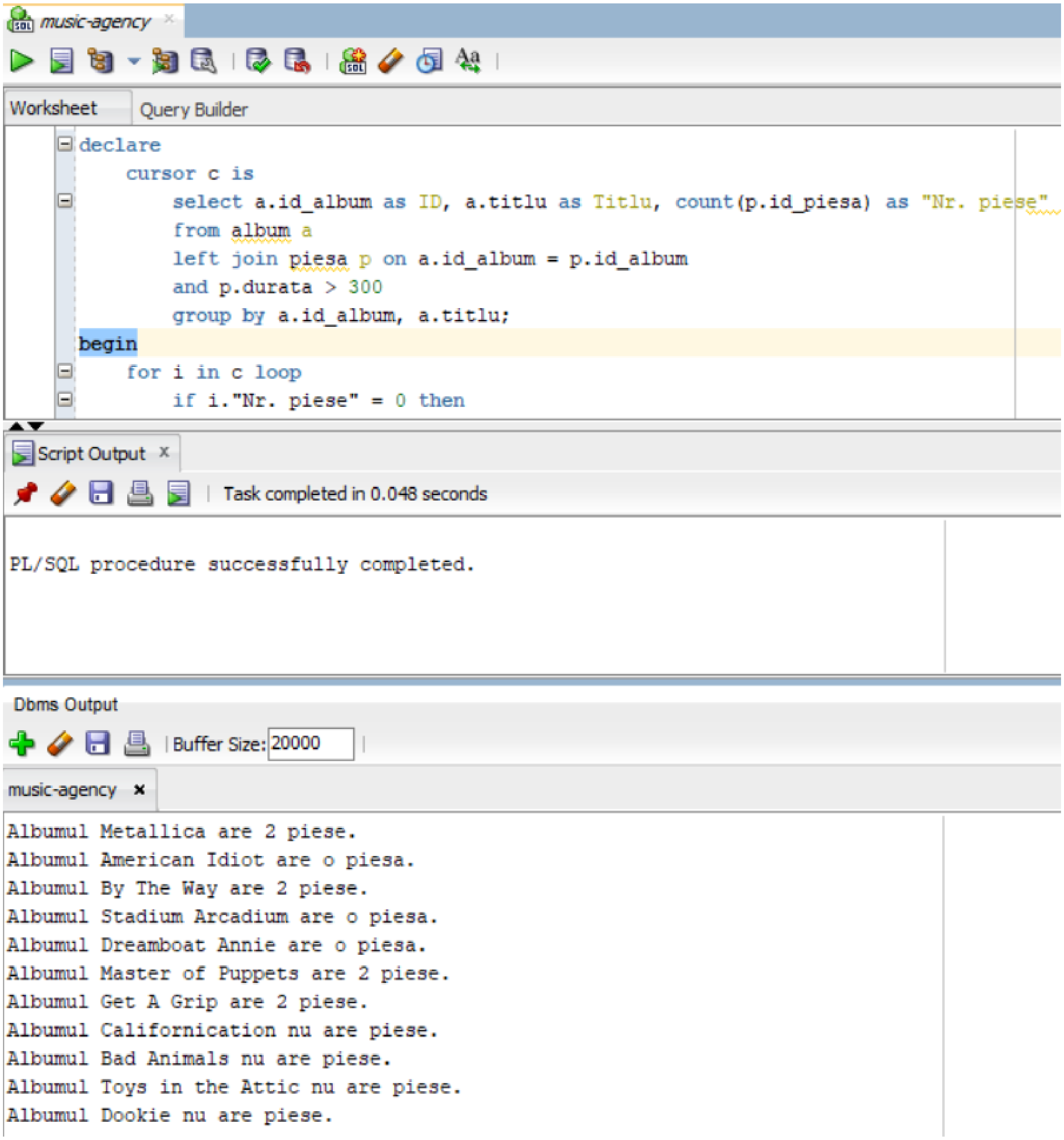
```
Managerul Anderson Rose gestioneaza o trupa.
Managerul Malcolm Hurst nu gestioneaza nici o trupa.
Managerul Kevin Henson nu gestioneaza nici o trupa.
Managerul Quinn Lester gestioneaza 2 trupe.
Managerul Roy Phillips gestioneaza o trupa.
Managerul Edward Austin gestioneaza o trupa.
```

CICLU CURSOR: in modelul de date personal, o PIESA apartine unui singur album, iar un ALBUM poate avea 1 sau mai multe piese. Fiecare PIESA are o anumita DURATA, specificata in secunde.

Am declarat un cursor pentru a retine numele fiecarui album si numarul de piese cu o durata de peste 300 de secunde. Pentru a parcurge liniile catre care pointeaza cursorul, am folosit un FOR cu o variabila “i”. Se va iesi automat din acest loop odata ce cursorul nu mai pointeaza catre nimic.

In functie de numarul de piese, am tratat 3 cazuri folosind un IF: daca albumul nu are piese, daca are o singura piesa sau daca are mai multe piese. Pentru fiecare caz se va afisa un mesaj corespunzator.

```
declare
    cursor c is
        select a.titlu as Titlu, count(p.id_piesa) as "Nr. piese"
        from album a
        left join piesa p on a.id_album = p.id_album
        and p.durata > 300
        group by a.id_album, a.titlu;
begin
    for i in c loop
        if i."Nr. piese" = 0 then
            dbms_output.put_line('Albumul ' || i.Titlu || ' nu are piese.');
        elsif i."Nr. piese" = 1 then
            dbms_output.put_line('Albumul ' || i.Titlu || ' are o piesa.');
        else
            dbms_output.put_line('Albumul ' || i.Titlu || ' are ' || i."Nr. piese" || ' piese.');
        end if;
    end loop;
end;
```



```
declare
    cursor c is
        select a.id_album as ID, a.titlu as Titlu, count(p.id_piesa) as "Nr. piese"
        from album a
        left join piesa p on a.id_album = p.id_album
        and p.durata > 300
        group by a.id_album, a.titlu;
begin
    for i in c loop
        if i."Nr. piese" = 0 then
```

Task completed in 0.048 seconds

```
PL/SQL procedure successfully completed.
```

Buffer Size: 20000

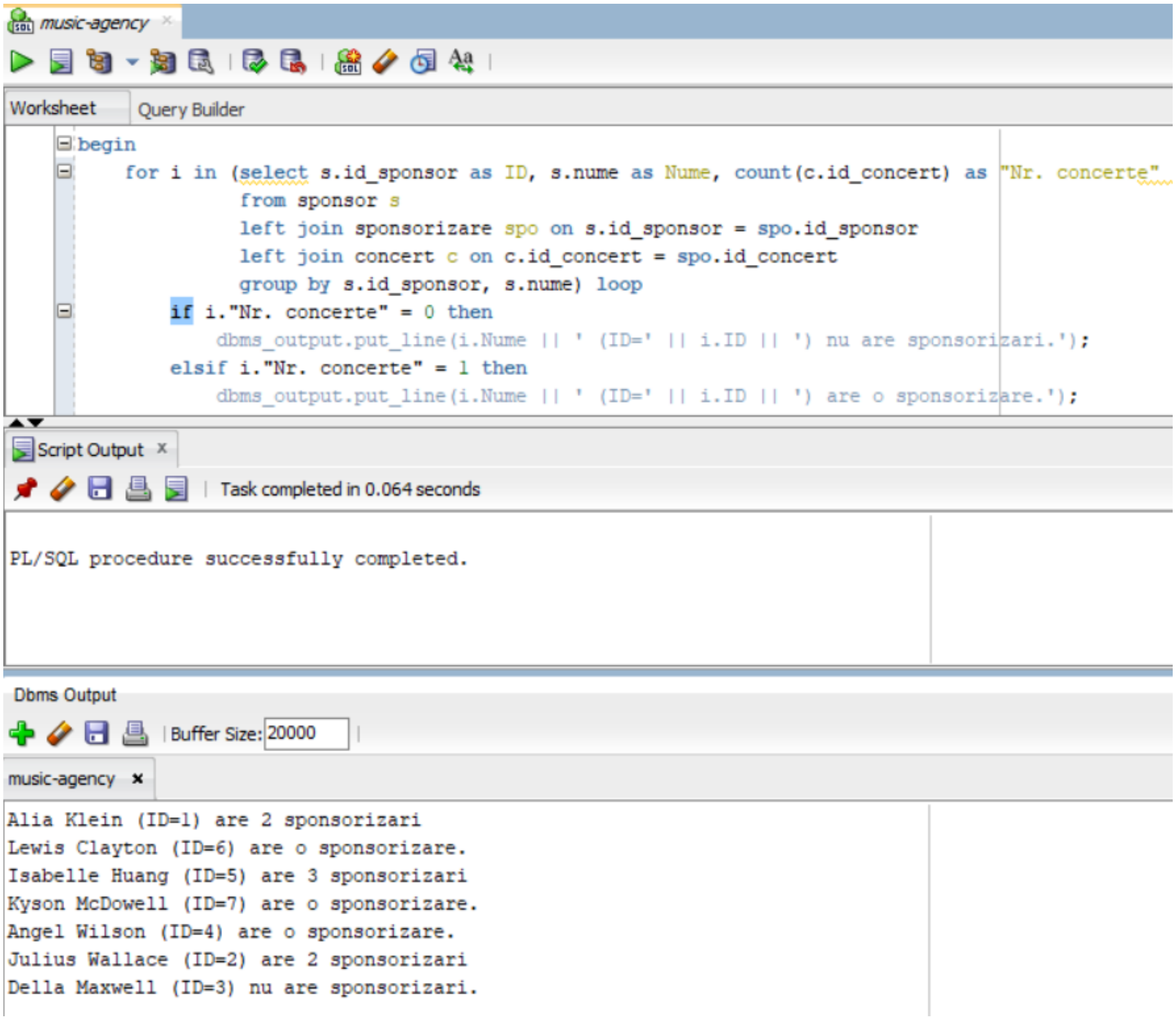
```
Albumul Metallica are 2 piese.
Albumul American Idiot are o piesa.
Albumul By The Way are 2 piese.
Albumul Stadium Arcadium are o piesa.
Albumul Dreamboat Annie are o piesa.
Albumul Master of Puppets are 2 piese.
Albumul Get A Grip are 2 piese.
Albumul Californication nu are piese.
Albumul Bad Animals nu are piese.
Albumul Toys in the Attic nu are piese.
Albumul Dookie nu are piese.
```

CICLU CURSOR CU SUBCERERI: in modelul de date personal, un CONCERT poate fi sponsorizat de 0 sau mai multi SPONSORI, iar un SPONSOR poate sponsoriza 0 sau mai multe CONCERTE.

In loc sa declar un cursor, am folosit un FOR pentru a parcurge direct liniile returnate de o subcerere. Aceasta subcerere selecteaza numarul de concerte sponsorizate pentru fiecare sponsor in parte.

In functie de numarul de sponsorizari, am tratat 3 cazuri folosind un IF: daca persoana respectiva nu are sponsorizari, daca sponsorizeaza un concert sau daca sponsorizeaza mai multe concerte. Pentru fiecare caz se va afisa un mesaj corespunzator.

```
begin
  for i in (select s.id_sponsor as ID, s.nume as Nume, count(c.id_concert) as "Nr. concerte"
       from sponsor s
       left join sponsorizare spo on s.id_sponsor = spo.id_sponsor
       left join concert c on c.id_concert = spo.id_concert
       group by s.id_sponsor, s.nume) loop
    if i."Nr. concerte" = 0 then
      dbms_output.put_line(i.Nume || ' (ID=' || i.ID || ') nu are sponsorizari.');
    elsif i."Nr. concerte" = 1 then
      dbms_output.put_line(i.Nume || ' (ID=' || i.ID || ') are o sponsorizare.');
    else
      dbms_output.put_line(i.Nume || ' (ID=' || i.ID || ') are ' || i."Nr. concerte" || ' sponsorizari');
    end if;
  end loop;
end;
```



```
begin
    for i in (select s.id_sponsor as ID, s.nume as Nume, count(c.id_concert) as "Nr. concerte"
              from sponsor s
              left join sponsorizare spo on s.id_sponsor = spo.id_sponsor
              left join concert c on c.id_concert = spo.id_concert
              group by s.id_sponsor, s.nume) loop
        if i."Nr. concerte" = 0 then
            dbms_output.put_line(i.Nume || ' (ID=' || i.ID || ') nu are sponsorizari.');
        elsif i."Nr. concerte" = 1 then
            dbms_output.put_line(i.Nume || ' (ID=' || i.ID || ') are o sponsorizare.');
```

Task completed in 0.064 seconds

```
PL/SQL procedure successfully completed.
```

```
Alia Klein (ID=1) are 2 sponsorizari
Lewis Clayton (ID=6) are o sponsorizare.
Isabelle Huang (ID=5) are 3 sponsorizari
Kyson McDowell (ID=7) are o sponsorizare.
Angel Wilson (ID=4) are o sponsorizare.
Julius Wallace (ID=2) are 2 sponsorizari
Della Maxwell (ID=3) nu are sponsorizari.
```

EXPRESII CURSOR: in modelul de date personal, CLIENTII semneaza CONTRACTE cu TRUPE, ca acele TRUPE sa cante la CONCERTELE CLIENTILOR. Un CONCERT este organizat de catre un singur CLIENT, iar un CLIENT poate organiza mai multe CONCERTE. La un CONCERT pot canta mai multe TRUPE, iar o TRUPA poate canta la mai multe CONCERTE. Un CONTRACT are cheia primara (X, Y, Z) => clientul X a semnat un contract ca trupa Y sa cante la concertul Z (concertul Z fiind organizat de catre clientul X). Un CLIENT poate semna mai multe CONTRACTE pentru acelasi CONCERT si poate semna mai multe CONTRACTE cu aceeasi TRUPA.

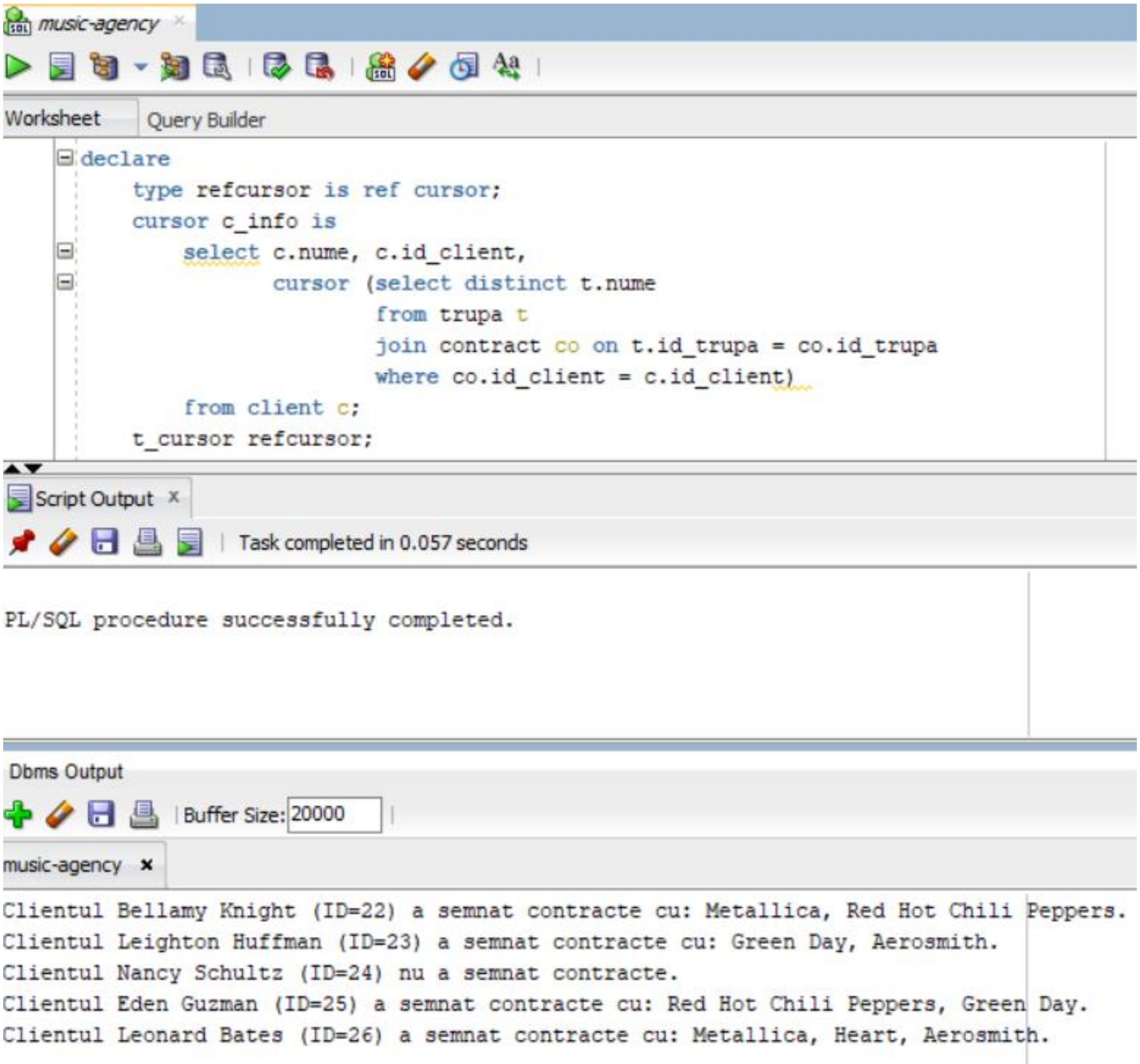
Pentru a retine numele unui client, ID-ul acestuia si trupele cu care a semnat contracte, am folosit expresii cursor. Am declarat un cursor care sa retina numele si ID-ul clientului, dar care sa retina si un alt cursor care pointeaza catre numele trupelor cu care a semnat contracte. Cursorul principal l-am utilizat ca la primul exercitiu: l-am deschis cu OPEN, am folosit un LOOP pentru a prelua liniile cu FETCH INTO (se iese din LOOP cand nu mai exista linii catre care sa pointeze cursorul) si l-am inchis la final cu CLOSE. Pentru celalalt subcursor, am operat similar: cu un LOOP (aceeasi conditie de exit) si FETCH INTO.

In functie de numarul de trupe cu care a semnat clientul respectiv contracte, am tratat 2 cazuri folosind un IF: daca a semnat contracte sau daca nu a semnat deloc.

```
declare
   type refcursor is ref cursor;
   cursor c_info is
      select c.nume, c.id_client,
           cursor (select distinct t.nume
                 from trupa t
                 join contract co on t.id_trupa = co.id_trupa
                 where co.id_client = c.id_client)
      from client c;
   t_cursor refcursor;
   c_nume client.nume%type;
   t_nume trupa.nume%type;
   c_id   client.id_client%type;
begin
   open c_info;
   loop
```

```
        fetch c_info into c_nume, c_id, t_cursor;
        exit when c_info%notfound;

        fetch t_cursor into t_nume;
        if t_cursor%notfound then
            dbms_output.put_line('Clientul ' || c_nume || ' (ID=' || c_id || ') nu a semnat contracte.');
        else
            dbms_output.put('Clientul ' || c_nume || ' (ID=' || c_id || ') a semnat contracte cu: ' || t_nume);
            loop
                fetch t_cursor into t_nume;
                exit when t_cursor%notfound;
                dbms_output.put(', ' || t_nume);
            end loop;
            dbms_output.put_line('.');
        end if;
    end loop;
    close c_info;
end;
```



```
declare
    type refcursor is ref cursor;
    cursor c_info is
        select c.nume, c.id_client,
               cursor (select distinct t.nume
                       from trupa t
                       join contract co on t.id_trupa = co.id_trupa
                       where co.id_client = c.id_client)
        from client c;
    t_cursor refcursor;
```

```
PL/SQL procedure successfully completed.
```

```
Clientul Bellamy Knight (ID=22) a semnat contracte cu: Metallica, Red Hot Chili Peppers.
Clientul Leighton Huffman (ID=23) a semnat contracte cu: Green Day, Aerosmith.
Clientul Nancy Schultz (ID=24) nu a semnat contracte.
Clientul Eden Guzman (ID=25) a semnat contracte cu: Red Hot Chili Peppers, Green Day.
Clientul Leonard Bates (ID=26) a semnat contracte cu: Metallica, Heart, Aerosmith.
```