1. CREAREA TABELELOR, INSERAREA DATELOR

```
CREATE TABLE trupa_mco (
   id_trupa NUMBER(4) PRIMARY KEY,
   nume VARCHAR(30) NOT NULL UNIQUE,
   data_infiintare DATE NOT NULL
);

CREATE TABLE membru_mco (
   id_membru NUMBER(4) PRIMARY KEY,
   id_trupa NUMBER(4) NOT NULL,
   nume VARCHAR(30) NOT NULL,
   rol VARCHAR(15) NOT NULL,
   FOREIGN KEY (id_trupa) REFERENCES trupa_mco(id_trupa)
);

CREATE TABLE client_mco (
   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
);

CREATE TABLE concert_mco (
   id_concert NUMBER(4) PRIMARY KEY,
   data DATE NOT NULL
);

CREATE TABLE contract_mco (
   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_mco(id_client),
    FOREIGN KEY (id_trupa) REFERENCES trupa_mco(id_trupa),
    FOREIGN KEY (id_concert) REFERENCES concert_mco(id_concert)
);

INSERT INTO trupa_mco
VALUES(1, 'Red Hot Chili Peppers', '15-NOV-1982');
INSERT INTO trupa_mco
VALUES(2, 'Metallica', '11-MAY-1981');
INSERT INTO trupa_mco
VALUES(3, 'Green Day', '26-JUN-1986');
INSERT INTO trupa_mco
VALUES(4, 'Heart', '2-FEB-1973');
INSERT INTO trupa_mco
VALUES(5, 'Aerosmith', '9-APR-1970');

INSERT INTO membru_mco
VALUES(1, 1, 'Michael Balzary', 'Basist');
INSERT INTO membru_mco
VALUES(2, 1, 'Anthony Kiedis', 'Vocalist');
INSERT INTO membru_mco
VALUES(3, 1, 'Chad Smith', 'Baterist');
INSERT INTO membru_mco
VALUES(4, 1, 'John Frusciante', 'Chitarist');
INSERT INTO membru_mco
VALUES(5, 2, 'James Hetfield', 'Vocalist');
INSERT INTO membru_mco
VALUES(6, 2, 'Kirk Hammett', 'Chitarist');
```

```
INSERT INTO membru_mco
VALUES(7, 2, 'Robert Trujillo', 'Basist');
INSERT INTO membru_mco
VALUES(8, 2, 'Lars Ulrich', 'Baterist');
INSERT INTO membru_mco
VALUES(9, 3, 'Billie Joe Armstrong', 'Vocalist');
INSERT INTO membru_mco
VALUES(10, 3, 'Michael Ryan Pritchard', 'Basist');
INSERT INTO membru_mco
VALUES(11, 3, 'Frank Edwin Wright III', 'Baterist');
INSERT INTO membru_mco
VALUES(12, 4, 'Ann Dustin Wilson', 'Vocalist');
INSERT INTO membru_mco
VALUES(13, 4, 'Nancy Lamoureux Wilson', 'Chitarist');
INSERT INTO membru_mco
VALUES(14, 4, 'Paul Moak', 'Pianist');
INSERT INTO membru_mco
VALUES(15, 4, 'Tony Lucido', 'Basist');
INSERT INTO membru_mco
VALUES(16, 4, 'Sean T. Lane', 'Baterist');
INSERT INTO membru_mco
VALUES(17, 4, 'Ryan Waters', 'Chitarist');
INSERT INTO membru_mco
VALUES(18, 4, 'Ryan Wariner', 'Chitarist');
INSERT INTO membru_mco
VALUES(19, 5, 'Steven Victor Tallarico', 'Vocalist');
INSERT INTO membru_mco
VALUES(20, 5, 'Thomas William Hamilton', 'Basist');
INSERT INTO membru_mco
VALUES(21, 5, 'Joseph Michael Kramer', 'Baterist');
INSERT INTO membru_mco
VALUES(22, 5, 'Burleigh Johnson', 'Pianist');
```

```
INSERT INTO membru_mco

VALUES(23, 5, 'Bradley Ernest Whitford', 'Chitarist');

INSERT INTO membru_mco

VALUES(24, 5, 'Joseph Anthony Pereira', 'Chitarist');


INSERT INTO client_mco

VALUES(22, 'Bellamy Knight', '+17167984922', 'bellamy.knight.480@gmail.com', '400 Commercial St, Medina, New York, 14103');

INSERT INTO client_mco

VALUES(23, 'Leighton Huffman', '+490821472510', 'leighton_huffman1329@outlook.com', 'Mellingburgredder 84, Happurg, Freistaat Bayern, Germany, 91230');

INSERT INTO client_mco

VALUES(24, 'Nancy Schultz', '+610246004601 ', 'nancyschultz319@gmail.com', '161 Keilor Rd, Essendon VIC 3040, Melbourne, Australia');

INSERT INTO client_mco

VALUES(25, 'Eden Guzman', '+330174277258', 'eden.guzman22@outlook.com', '76 Rue des Belles Feuilles, 75116, Paris, France');

INSERT INTO client_mco

VALUES(26, 'Leonard Bates', '+81997395728', 'leonard.bates09@gmail.com', 'Senkawa-dori Avenue, 176-0006, Tokyo, Japan');


INSERT INTO concert_mco

VALUES(100, '05-AUG-2024');

INSERT INTO concert_mco

VALUES(110, '28-DEC-2024');

INSERT INTO concert_mco

VALUES(120, '12-OCT-2024');

INSERT INTO concert_mco

VALUES(130, '02-NOV-2024');

INSERT INTO concert_mco

VALUES(140, '14-FEB-2025');


INSERT INTO contract_mco

VALUES(22, 1, 100, '01-JUN-2024', 12500);
```

```
INSERT INTO contract_mco (id_client, id_trupa, id_concert, data_semnare)
VALUES(22, 2, 100, '14-JUL-2024');
INSERT INTO contract_mco
VALUES(26, 4, 140, '01-SEP-2024', 15000);
INSERT INTO contract_mco
VALUES(23, 3, 110, '08-DEC-2024', 14000);
INSERT INTO contract_mco
VALUES(25, 1, 130, '25-OCT-2024', 10000);
INSERT INTO contract_mco
VALUES(26, 5, 140, '25-OCT-2024', 17500);
INSERT INTO contract_mco (id_client, id_trupa, id_concert, data_semnare)
VALUES(23, 5, 120, '10-SEP-2024');
INSERT INTO contract_mco
VALUES(23, 5, 110, '03-AUG-2024', 9000);
INSERT INTO contract_mco
VALUES(26, 2, 140, '07-DEC-2024', 13500);
INSERT INTO contract_mco
VALUES(25, 3, 130, '19-AUG-2024', 13500);
COMMIT;

CREATE TABLE contract_aux AS (SELECT * FROM contract_mco);
```

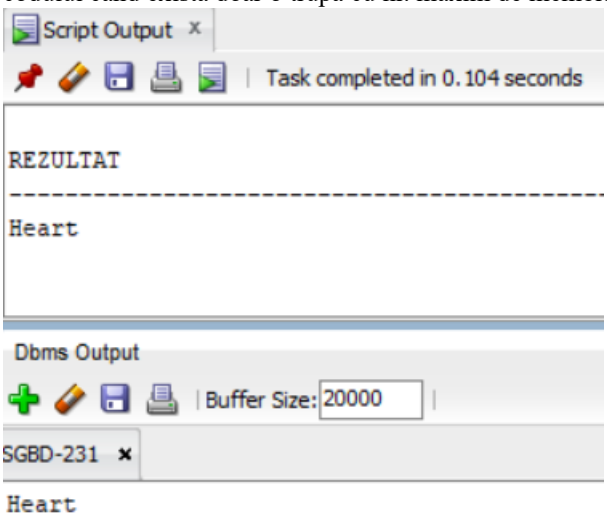
EXERCITIUL 5 ADAPTAT

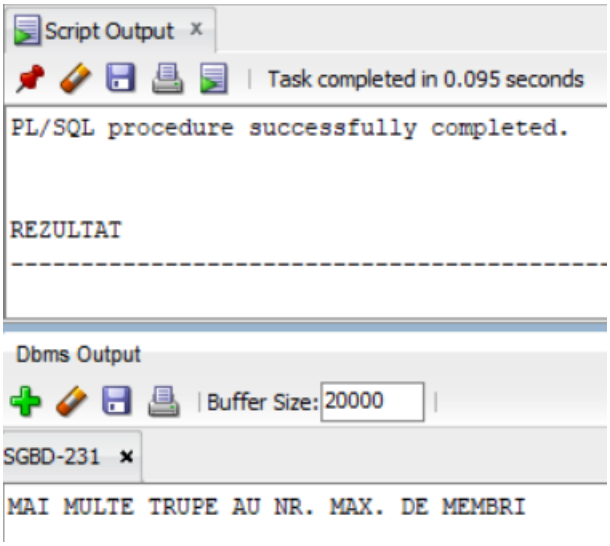
1. Cerinta: Definiti un bloc anonim prin care sa se afle numele trupei cu cei mai multi membri. Comentati cazul in care exista cel putin doua trupe cu numar maxim de membri. Problema se va rezolva utilizand variabile de legatura; afisati rezultatul atat din bloc, cat si din exteriorul acestuia.
2. Rezolvare:

```
variable rezultat varchar2;
variable max_membri number;
begin
   select max(count(1))
   into :max_membri
   from membru_mco
```

```
    group by id_trupa;


    select t.nume
    into :rezultat
    from trupa_mco t
    join membru_mco m on t.id_trupa = m.id_trupa
    group by t.nume
    having count(1) = :max_membri;


    dbms_output.put_line(:rezultat);


    exception
        when too_many_rows then
            dbms_output.put_line('MAI MULTE TRUPE AU NR. MAX. DE MEMBRI');
            :rezultat := null;


end;
/
print rezultat;
```

3. Explicatii si poze: am utilizat o variabila de legatura in care am stocat numarul maxim de membri dintr-o trupa, cu un select simplu. Apoi, am selectat numele trupei care are numarul de membri egal cu numarul calculat mai devreme, stocand rezultatul in alta variabila de legatura. Am tratat exceptia (in cazul in care exista mai mult de o trupa cu nr. maxim de membri) si am afisat rezultatul. Am inclus 2 poze: rularea codului cand exista doar o trupa cu nr. maxim de membri, si rularea codului cand exista 2 sau mai multe.





EXERCITIUL 7 ADAPTAT:

1. Cerinta: Determinati numarul de contracte pentru un client al carui cod este dat de la tastatura, si onorariile pe care acesta le plateste trupelor. Acest client va face parte dintr-o categorie:
   - categoria A: 0 <= onorarii < 10000
   - categoria B: 10000 <= onorarii < 20000
   - categoria C: 20000 <= onorarii < 30000
   - categoria D: onorarii >= 30000

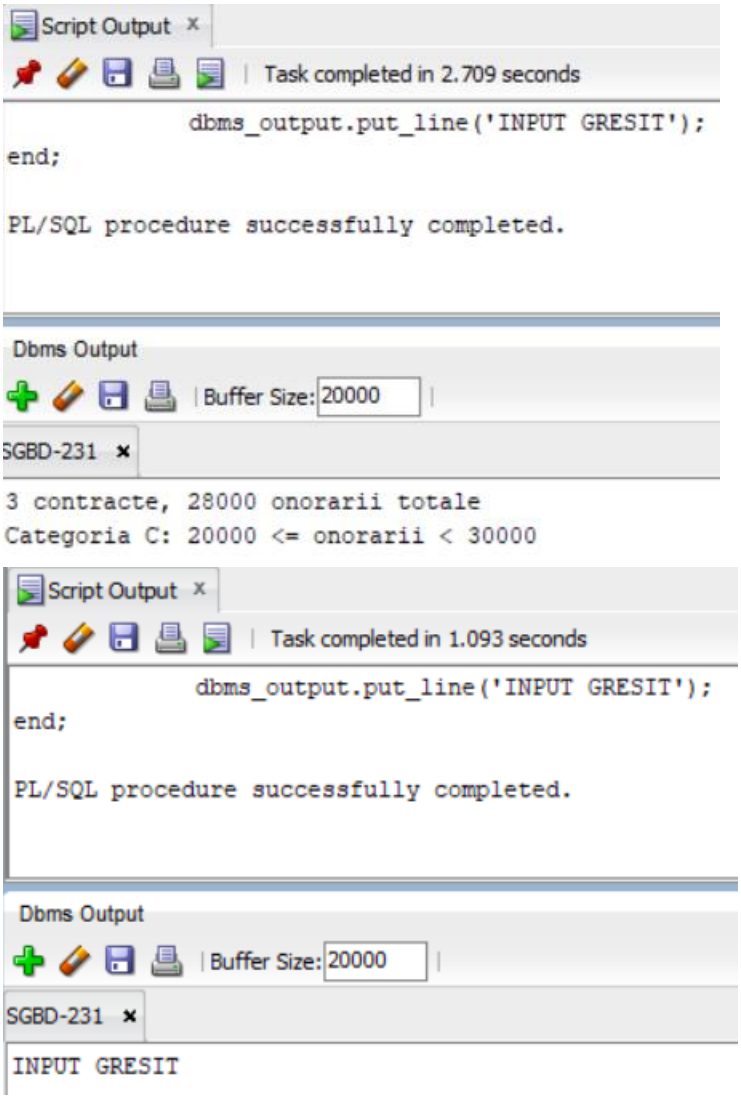
   Afisati numarul de contracte, onorariile platite si categoria. Sa se comenteze cazul in care nu exista un client cu codul specificat.

2. Rezolvare:

```
declare
   cl_cod           client_mco.id_client%type := &cod;
   cl_nr_contracte   number                   := 0;
   cl_onorariu_total number                   := 0;
begin
      select id_client
      into cl_cod
      from client_mco
      where id_client = cl_cod;

      select count(1), coalesce(sum(onorariu), 0)
      into cl_nr_contracte, cl_onorariu_total
```

```
    from contract_mco
    where id_client = cl_cod;

    dbms_output.put_line(cl_nr_contracte || ' contracte, ' || cl_onorariu_total || ' onorarii totale');

    if cl_onorariu_total < 10000 then
       dbms_output.put_line('Categoria A: 0 <= onorarii < 10000');
    elsif cl_onorariu_total < 20000 then
       dbms_output.put_line('Categoria B: 10000 <= onorarii < 20000');
    elsif cl_onorariu_total < 30000 then
       dbms_output.put_line('Categoria C: 20000 <= onorarii < 30000');
    else
       dbms_output.put_line('Categoria D: onorarii > 30000');
    end if;

  exception
    when no_data_found then
        dbms_output.put_line('INPUT GRESIT');
end;
```

3. Explicatii si poze: In primul rand, am verificat (prin primul select) daca exista un client avand codul dat. In cazul in care nu exista, se va trece pe exceptia tratata. In urmatorul select am facut rost de numarul de contracte si de onorariile din fiecare contract adunate pentru clientul specific, stocand cele 2 rezultate in 2 variabile. Am afisat pe o linie numarul de contracte si onorariile calculate. Apoi, l-am incadrat pe client intr-o categorie in functie de onorariile platite, utilizand instructiunea IF, afisand categoria pe urmatoarea linie.
   Inputul pe care l-am dat: 23 (3 contracte, 28000 onorarii, categoria C), 21 (nu exista).



EXERCITIUL 9 ADAPTAT:

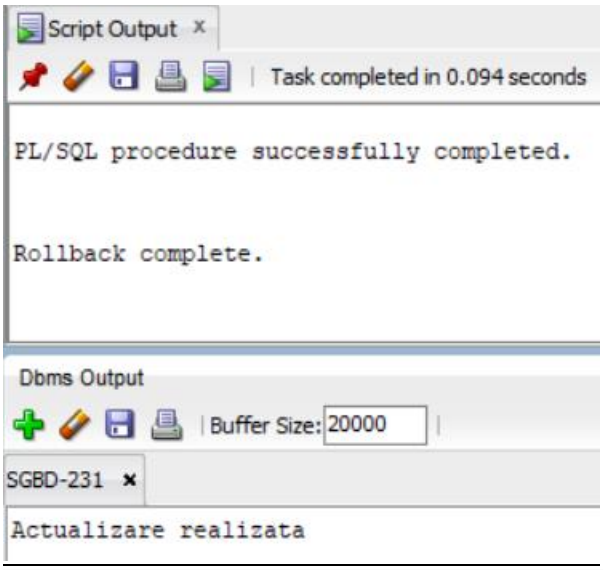
1. Cerinta: Scrieti un bloc PL/SQL in care sa stocati prin variabile de substitutie un cod de client si procentul cu care acesta trebuie sa mareasca onorariul platit pentru fiecare contract pe care il are. Daca modificarea s-a putut realiza (exista in tabelul CLIENT_MCO un client avand codul respectiv) sa se afiseze mesajul "Actualizare realizata", iar in caz contrar mesajul "Nu exista un angajat cu acest cod". Anulati modificarile realizate.

2. Rezolvare:

```
define p_id_client    = 23
define p_procent      = 10
declare
   t_id_client contract_aux.id_client%type := &p_id_client;
```

```
    t_procent       number              := &p_procent;
begin
    select id_client
    into t_id_client
    from client_mco
    where id_client = t_id_client;

    update contract_aux
    set onorariu = onorariu + (t_procent / 100) * onorariu
    where id_client = t_id_client;

    dbms_output.put_line('Actualizare realizata');

    exception
        when no_data_found then
            dbms_output.put_line('Nu exista un angajat cu acest cod');
end;
/
rollback;
```

3. Explicatii si poze: mai intai am definit input-ul pentru codul de client si procentul mentionat, apoi le-am stocat in 2 variabile. Am verificat daca exista clientul cu codul dat (prin primul select si exceptia de la final), apoi am folosit comanda de update ca sa maresc onorariile de platit si am afisat un mesaj corespunzator. Am inclus 4 poze (una in care clientul nu exista, si 3 in care s-a realizat modificarea si am aratat before-and-after).



	ID_CLIENT	ID_TRUPA	ID_CONCERT	DATA_SEMNARE	ONORARIU
1	22	1	100	01-JUN-24	12500
2	22	2	100	14-JUL-24	5000
3	26	4	140	01-SEP-24	15000
4	23	3	110	08-DEC-24	14000
5	25	1	130	25-OCT-24	10000
6	26	5	140	25-OCT-24	17500
7	23	5	120	10-SEP-24	5000
8	23	5	110	03-AUG-24	9000
9	26	2	140	07-DEC-24	13500
10	25	3	130	19-AUG-24	13500

	ID_CLIENT	ID_TRUPA	ID_CONCERT	DATA_SEMNARE	ONORARIU
1	22	1	100	01-JUN-24	12500
2	22	2	100	14-JUL-24	5000
3	26	4	140	01-SEP-24	15000
4	23	3	110	08-DEC-24	15400
5	25	1	130	25-OCT-24	10000
6	26	5	140	25-OCT-24	17500
7	23	5	120	10-SEP-24	5500
8	23	5	110	03-AUG-24	9900
9	26	2	140	07-DEC-24	13500
10	25	3	130	19-AUG-24	13500

Script Output × Query Result ×

Task completed in 0.444 seconds

```
PL/SQL procedure successfully completed.


Rollback complete.
```

Dbms Output

Buffer Size: 20000

SGBD-231 ×

```
Nu exista un angajat cu acest cod
```