

Tema 3 - Laborator

Aelenei Alex

Cuprins

1	Introducere	2
2	Exercițiul 9	2
3	Exercițiul 10	4

1 Introducere

Diagrama conceptuală și diagrama entitate-relație utilizate în această temă sunt mai jos. Datele de test folosite sunt cele utilizate în tema 1 și cele de la finalul acestui document.

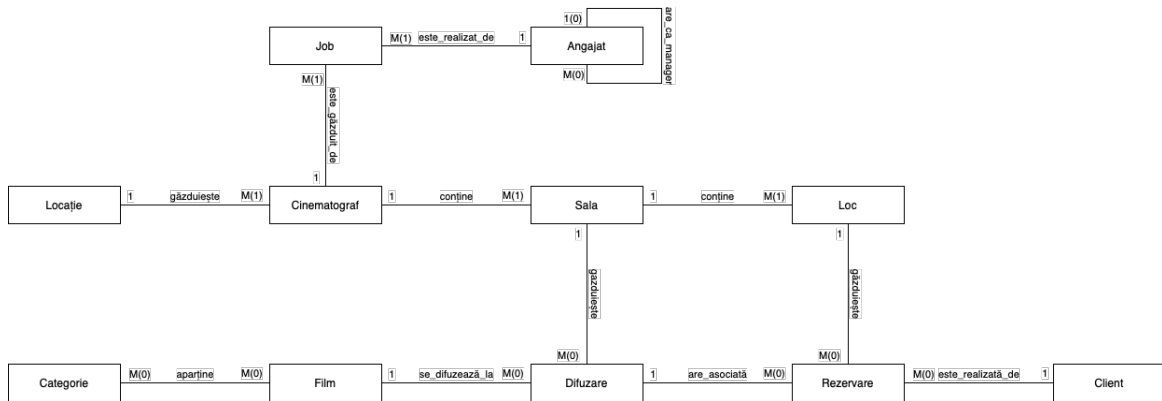


Figura 1: Diagrama conceptuală

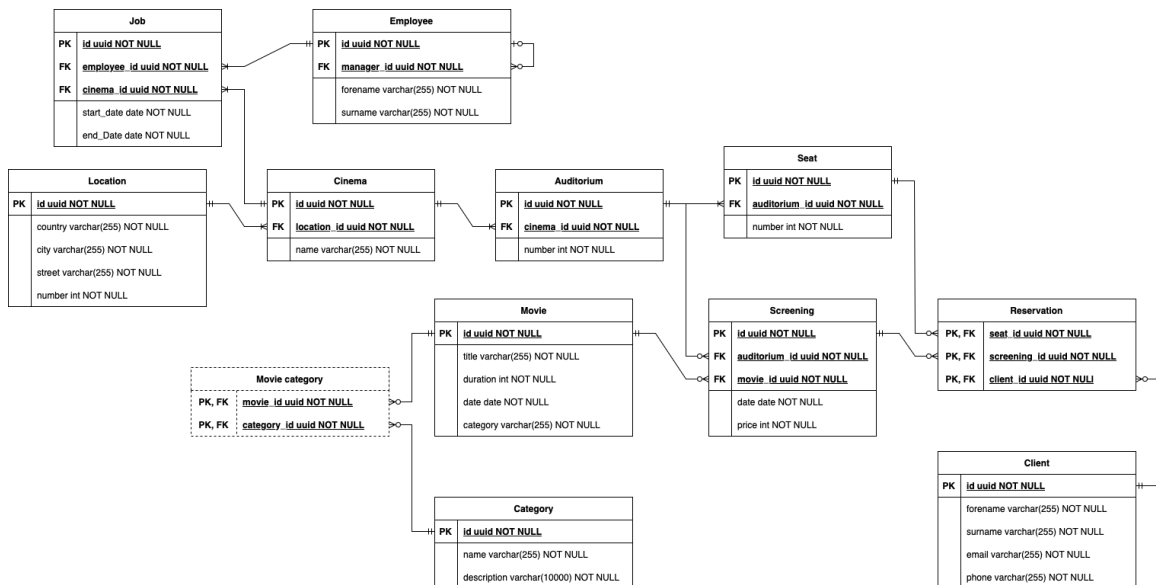


Figura 2: Diagrama entitate-relație

Valorile utilizate pentru a testa query-urile sunt cele citate în Tema 2. De asemenea, spre deosebire de Tema 1 și Tema 2, aceste query-uri au fost rulate pe o instanță de Oracle Database 19, bazată pe imaginea oficială de la Oracle, dar cu un build pentru MacOS ARM local.

2 Exercițiul 9

Problema 9 din *Laborator2_PLSQL.pdf* spune: *Definiți tipul subordonati_*** (vector, dimensiune maximă 10, menține numere). Creați tabelul manageri_*** cu următoarele câmpuri: cod_mgr NUMBER(10), nume VARCHAR2(20), lista_subordonati_***. Introduceți 3 linii în tabel. Afișați informațiile din tabel. Ștergeți tabelul creat, apoi tipul. Adaptând cerința la structura utilizată vom rezolva următoarea cerință: Determinați clienții care au același nume de familie, reținând prenumele lor într-un tip de date (vector, reține cel mult 10 nume).*

```
01 | SET VERIFY OFF
02 |
03 | SET FEEDBACK ON
04 |
05 | SET SERVEROUTPUT ON
```

```

06 |
07 | CREATE OR REPLACE TYPE NAME_LIST AS
08 |     VARRAY(10) OF VARCHAR2(255);
09 | /
10 |
11 | CREATE TABLE SHARE_LAST_NAME (
12 |     LAST_NAME VARCHAR2(255),
13 |     FIRST_NAMES NAME_LIST
14 | );
15 |
16 | DECLARE
17 |     NAMES NAME_LIST;
18 | BEGIN
19 |     FOR ITEM IN (
20 |         SELECT
21 |             C1.LAST_NAME
22 |         FROM
23 |             CLIENT C1
24 |             INNER JOIN CLIENT C2
25 |                 ON UPPER(C1.LAST_NAME) = UPPER(C2.LAST_NAME)
26 |                 AND C1.CLIENT_ID < C2.CLIENT_ID
27 |         GROUP BY
28 |             C1.LAST_NAME
29 |     ) LOOP
30 |         SELECT
31 |             UPPER(C.FIRST_NAME) BULK COLLECT INTO NAMES
32 |         FROM
33 |             CLIENT C
34 |         WHERE
35 |             UPPER(C.LAST_NAME) = UPPER(ITEM.LAST_NAME);
36 |         INSERT INTO SHARE_LAST_NAME VALUES (
37 |             UPPER(ITEM.LAST_NAME),
38 |             NAMES
39 |         );
40 |     END LOOP;
41 |
42 |     FOR ITEM IN (
43 |         SELECT
44 |             *
45 |         FROM
46 |             SHARE_LAST_NAME
47 |     ) LOOP
48 |         DBMS_OUTPUT.PUT_LINE(ITEM.LAST_NAME
49 |                               || ':');
50 |         FOR VALUE_INDEX IN ITEM.FIRST_NAMES.FIRST..ITEM.FIRST_NAMES.LAST LOOP
51 |             DBMS_OUTPUT.PUT_LINE(ITEM.FIRST_NAMES(VALUE_INDEX));
52 |         END LOOP;
53 |
54 |         DBMS_OUTPUT.PUT_LINE('');
55 |     END LOOP;
56 | END;
57 | /
58 |
59 | DROP TABLE SHARE_LAST_NAME;
60 |
61 | DROP TYPE NAME_LIST;

```

Listing 1: Query pentru exercițiul 9

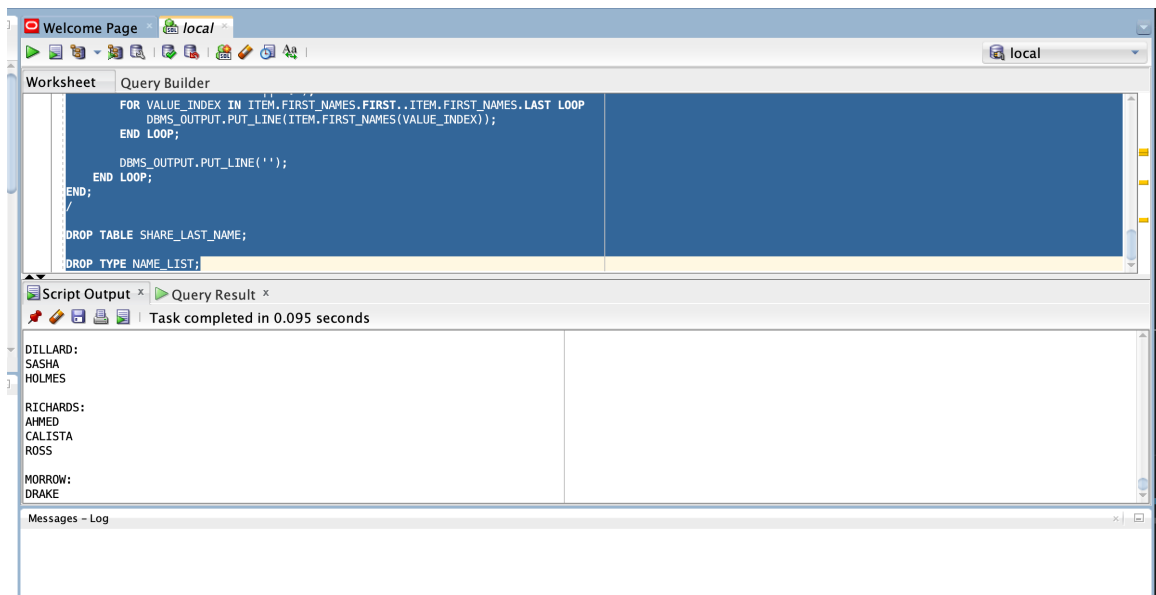


Figura 3: Rezultat query exercițiul 9

3 Exercițiul 10

Problema 10 din *Laborator2_PLSQL.pdf* spune: Creați tabelul `emp_test_***` cu coloanele `employee_id` și `last_name` din tabelul `employees`. Adăugați în acest tabel un nou câmp numit telefon de tip tablou imbricat. Acest tablou va menține pentru fiecare salariat toate numerele de telefon la care poate fi contactat. Inserați o linie nouă în tabel. Actualizați o linie din tabel. Afișați informațiile din tabel. Ștergeți tabelul și tipul. Adaptând cerința la structura utilizată vom rezolva următoarea cerință: Generați tabelul `CLIENT_SUMMARY`, care conține trei coloane: ID-ul clientului, numele complet și o coloana de tip tabel imbricat care conține datele de contact ale utilizatorului.

```

01 | SET VERIFY OFF
02 |
03 | SET FEEDBACK ON
04 |
05 | SET SERVEROUTPUT ON
06 |
07 | CREATE TABLE CLIENT_SUMMARY AS
08 |     SELECT
09 |         C.CLIENT_ID,
10 |         UPPER(C.FIRST_NAME
11 |             || ' '
12 |             || C.LAST_NAME) AS FULL_NAME
13 |     FROM
14 |         CLIENT C;
15 |
16 | /
17 |
18 | CREATE OR REPLACE TYPE CONTACT_DATA IS
19 |     TABLE OF VARCHAR2(255);
20 | /
21 |
22 | ALTER TABLE CLIENT_SUMMARY
23 |     ADD (
24 |         CONTACT CONTACT_DATA
25 |     ) NESTED TABLE CONTACT STORE AS CONTACT_DATA_SUMMARY;
26 |
27 | /
28 |
29 | DECLARE
30 |     CONTACTS     CONTACT_DATA;
31 |     OUTPUT_LINE  VARCHAR2(1000);
32 | BEGIN
33 |     FOR ITEM IN (
34 |         SELECT
35 |             *
36 |         FROM
37 |             CLIENT

```

```

38 |     ) LOOP
39 |         UPDATE CLIENT_SUMMARY CS
40 |         SET
41 |             CONTACT = CONTACT_DATA(
42 |                 ITEM.PHONE_NUMBER,
43 |                 ITEM.EMAIL
44 |             )
45 |         WHERE
46 |             CS.CLIENT_ID = ITEM.CLIENT_ID;
47 |     END LOOP;
48 |
49 |     FOR ITEM IN (
50 |         SELECT
51 |             *
52 |         FROM
53 |             CLIENT_SUMMARY
54 |     ) LOOP
55 |         DBMS_OUTPUT.PUT_LINE('CLIENT_ID: '
56 |                               || ITEM.CLIENT_ID);
57 |         DBMS_OUTPUT.PUT_LINE('FULL_NAME: '
58 |                               || ITEM.FULL_NAME);
59 |         OUTPUT_LINE := '';
60 |         FOR CONTACT_INDEX IN ITEM.CONTACT.FIRST..ITEM.CONTACT.LAST LOOP
61 |             OUTPUT_LINE := OUTPUT_LINE
62 |                             || ITEM.CONTACT(CONTACT_INDEX)
63 |                             || ' ';
64 |         END LOOP;
65 |
66 |         DBMS_OUTPUT.PUT_LINE('CONTACT: '
67 |                               || OUTPUT_LINE);
68 |         DBMS_OUTPUT.PUT_LINE(' ');
69 |     END LOOP;
70 | END;
71 | /
72 |
73 | DROP TABLE CLIENT_SUMMARY;
74 |
75 | DROP TYPE CONTACT_DATA;

```

Listing 2: Query pentru exercițiul 10

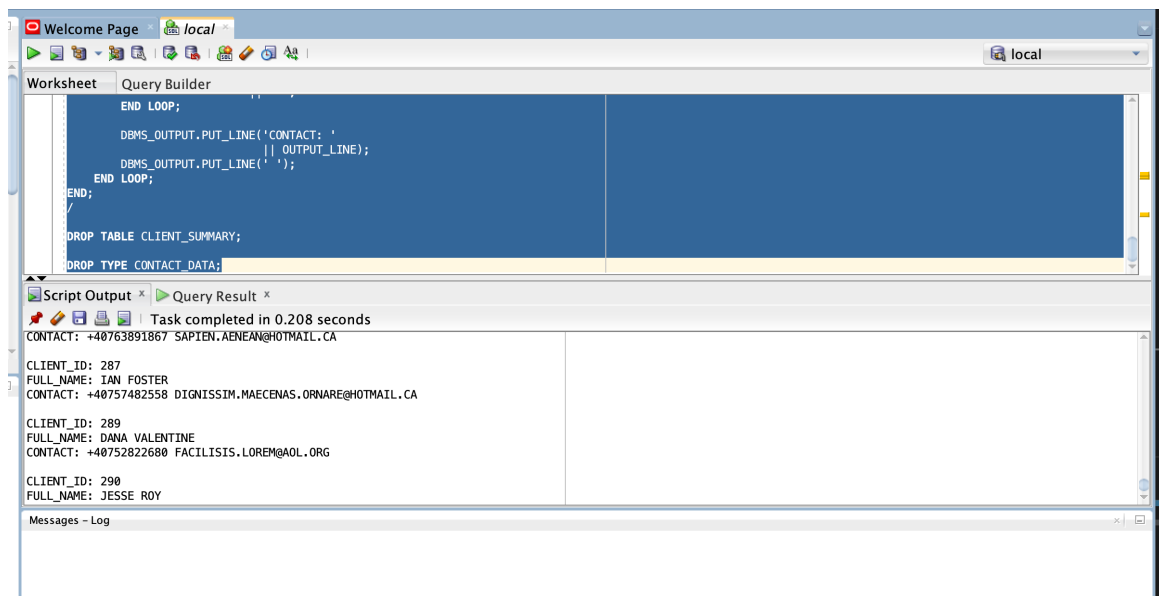


Figura 4: Rezultat query exercițiul 10