

Gestionarea unui lanț hotelier

Bouruc Petru-Liviu

09/01/2021

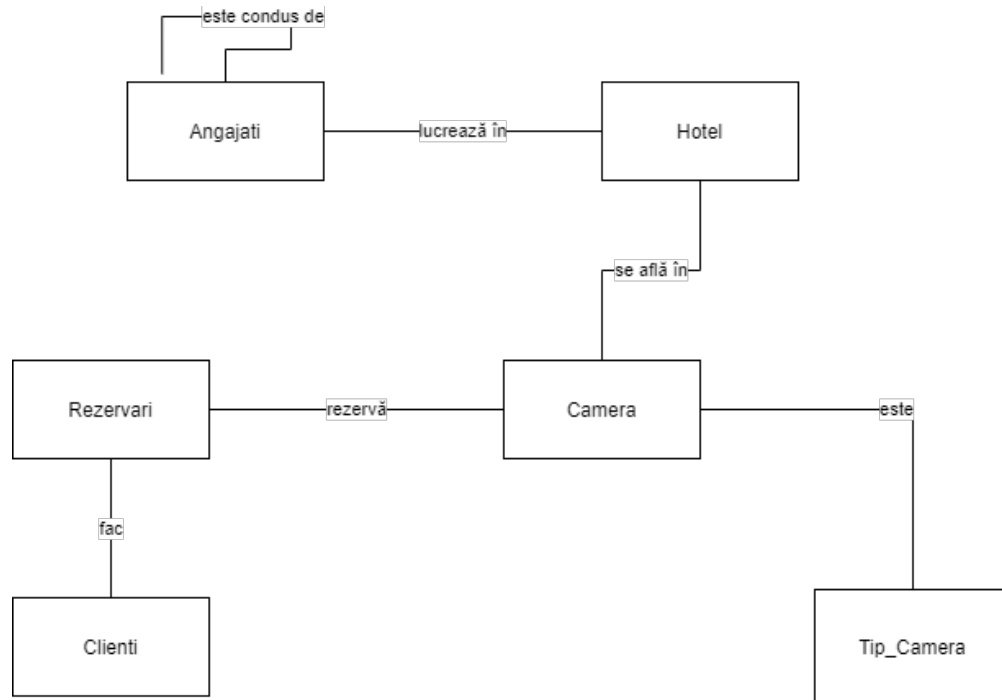
Contents

1	Introducere & Descrierea proiectului	2
2	Diagrama Entitate-Relatie	2
3	Diagrama conceptuala	3
4	Crearea bazei de date	3
5	Popularea bazei de date	5
6	Cerința 6	6
7	Cerința 7	7
8	Cerința 8	8
9	Cerința 9	11
10	Cerința 10	14
11	Cerința 11	15
12	Cerința 12	16
13	Cerința 13	17
14	Cerința 14	20

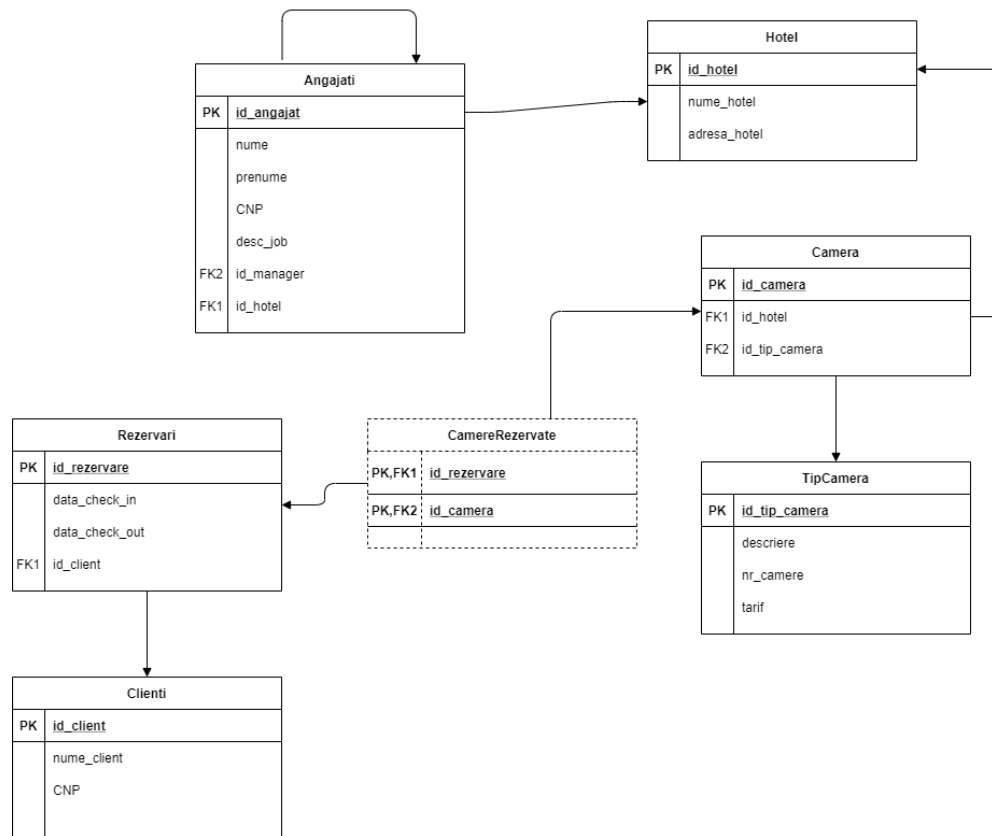
1 Introducere & Descrierea proiectului

Pentru acest proiect am ales să realizez gestionarea unui lanț hotelier. Astfel, voi cuprinde date precum diverse hoteluri, angajații pentru fiecare dintre acestea, cât și de ce camere dispun. Totodată, există posibilitatea de a crea rezervări pentru camere, un client putând realiza mai multe rezervări. O rezervare poate fi pentru o cameră sau pentru mai multe.

2 Diagrama Entitate-Relatie



3 Diagrama conceptuala



4 Crearea bazei de date

Începând de la această secțiune, voi atașa codul aferent fiecărei cerințe, alături de un print-screen care dovedește corectitudinea codului.

```
CREATE TABLE Hotel(
    id_hotel NUMBER PRIMARY KEY,
    nume_hotel VARCHAR2(100),
    adresa_hotel VARCHAR2(100));
```

```
CREATE TABLE Angajati (
    id_angajat NUMBER PRIMARY KEY,
    nume VARCHAR2(100),
    prenume VARCHAR2(100),
    CNP NUMBER,
    desc_job VARCHAR2(100),
    id_manager NUMBER,
    id_hotel NUMBER,
    FOREIGN KEY (id_hotel) REFERENCES Hotel(id_hotel) ON DELETE SET NULL,
    FOREIGN KEY (id_manager) REFERENCES Angajati(id_angajat) ON DELETE SET NULL);
```

```
CREATE TABLE TipCamera(
    id_tip_camera NUMBER PRIMARY KEY,
    descriere VARCHAR2(100),
```

```
nr_camere NUMBER NOT NULL,
tarif NUMBER);
```

```
CREATE TABLE Camera(
    id_camera NUMBER PRIMARY KEY,
    id_hotel NUMBER,
    id_tip_camera NUMBER,
    nr_camera NUMBER,
    FOREIGN KEY (id_hotel) REFERENCES Hotel(id_hotel) ON DELETE SET NULL,
    FOREIGN KEY (id_tip_camera) REFERENCES TipCamera(id_tip_camera) ON DELETE SET NULL);
```

```
CREATE TABLE Clienti(
    id_client NUMBER PRIMARY KEY,
    nume_client VARCHAR2(100),
    CNP NUMBER);
```

```
CREATE TABLE Rezervari(
    id_rezervare NUMBER PRIMARY KEY,
    data_check_in DATE,
    data_check_out DATE,
    id_client NUMBER,
    FOREIGN KEY (id_client) REFERENCES Clienti(id_client) ON DELETE SET NULL);
```

```
CREATE TABLE CamereRezervate(
    id_camera NUMBER,
    id_rezervare NUMBER,
    PRIMARY KEY (id_camera, id_rezervare),
    FOREIGN KEY (id_camera) REFERENCES Camera(id_camera) ON DELETE SET NULL,
    FOREIGN KEY (id_rezervare) REFERENCES Rezervari(id_rezervare) ON DELETE SET NULL);
```

Live SQL
Feedback
Help
bouruc.livu2000@gmail.com

SQL Worksheet

Clear
Find
Actions
Save
Run

```

1 CREATE TABLE Hotel(
2   id_hotel NUMBER PRIMARY KEY,
3   nume_hotel VARCHAR2(100),
4   adresa_hotel VARCHAR2(100));
5
6 CREATE TABLE Angajati (
7   id_angajat NUMBER PRIMARY KEY,
8   nume VARCHAR2(100),
9   prenume VARCHAR2(100),
10  CNP NUMBER,
11  desc_job VARCHAR2(100),
12  id_manager NUMBER,
13  id_hotel NUMBER,
14  FOREIGN KEY (id_hotel) REFERENCES Hotel(id_hotel) ON DELETE SET NULL,
15  FOREIGN KEY (id_manager) REFERENCES Angajati(id_angajat) ON DELETE SET NULL);
16
17 CREATE TABLE TipCamera(
18   id_tip_camera NUMBER PRIMARY KEY,
19   descriere VARCHAR2(100),
20   nr_camere NUMBER NOT NULL,
21   tarif NUMBER);

```

Table created.
 Table created.
 Table created.
 Table created.
 Table created.

5 Popularea bazei de date

```
INSERT INTO Hotel VALUES(0, 'Racova', 'Str. Stefan cel Mare, nr.10, Vaslui');
INSERT INTO Hotel VALUES(1, 'Europa', 'Str. Stefan cel Mare, nr.11, Vaslui');
INSERT INTO Hotel VALUES(2, 'Intercontinental', 'Bd. Elisabeta, Bucuresti');

INSERT INTO Angajati VALUES(0, 'Liviu', 'Bouruc', 123456, 'Director', NULL, 2);
INSERT INTO Angajati VALUES(1, 'Miruna', 'Vasilu', 123457, 'Manager', 0, 2);
INSERT INTO Angajati VALUES(2, 'Ciuri', 'Ciurescu', 123458, 'Bucatar-Sef', 0, 2);
INSERT INTO Angajati VALUES(3, 'Mihai', 'Dobrescu', 123459, 'Receptioner', 1, 2);
INSERT INTO Angajati VALUES(4, 'David', 'Popa', 123410, 'Bucatar', 2, 2);
INSERT INTO Angajati VALUES(5, 'Andrei', 'Filip', 123333, 'CEO', NULL, 0);
INSERT INTO Angajati VALUES(6, 'Marius', 'Dumi', 123334, 'Manager', NULL, 1);
INSERT INTO Angajati VALUES(7, 'Tode', 'Lavric', 1233555, 'Majordom', 6, 1);

INSERT INTO TipCamera VALUES(0, 'Camera Single', 1, 120);
INSERT INTO TipCamera VALUES(1, 'Apartament', 2, 200);
INSERT INTO TipCamera VALUES(2, 'Penthouse', 4, 500);

INSERT INTO Camera VALUES(0, 2, 0, 11);
INSERT INTO Camera VALUES(1, 2, 0, 12);
INSERT INTO Camera VALUES(2, 2, 1, 21);
INSERT INTO Camera VALUES(3, 2, 2, 22);
INSERT INTO Camera VALUES(4, 2, 2, 30);
INSERT INTO Camera VALUES(5, 1, 0, 11);
INSERT INTO Camera VALUES(6, 1, 0, 12);
INSERT INTO Camera VALUES(7, 1, 1, 13);
INSERT INTO Camera VALUES(8, 0, 0, 11);
INSERT INTO Camera VALUES(9, 0, 0, 12);

INSERT INTO Clienti VALUES(0, 'George', 222222);
INSERT INTO Clienti VALUES(1, 'Laur', 22223);
INSERT INTO Clienti VALUES(2, 'Stafie', 233344);
INSERT INTO Clienti VALUES(3, 'Lila', 233345);
INSERT INTO Clienti VALUES(4, 'Costea', 233349);

INSERT INTO Rezervari VALUES(0, TO_DATE('2017-03-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'),
↪ TO_DATE('2017-04-30 19:35:20', 'YYYY-MM-DD HH24:MI:SS'), 0);
INSERT INTO Rezervari VALUES(1, TO_DATE('2018-06-2 19:35:20', 'YYYY-MM-DD HH24:MI:SS'),
↪ TO_DATE('2018-07-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'), 1);
INSERT INTO Rezervari VALUES(2, TO_DATE('2019-03-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'),
↪ TO_DATE('2019-05-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'), 1);
INSERT INTO Rezervari VALUES(3, TO_DATE('2017-03-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'),
↪ TO_DATE('2017-08-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'), 2);
INSERT INTO Rezervari VALUES(4, TO_DATE('2017-03-1 19:35:20', 'YYYY-MM-DD HH24:MI:SS'),
↪ TO_DATE('2017-03-31 19:35:20', 'YYYY-MM-DD HH24:MI:SS'), 3);

INSERT INTO CamereRezervate VALUES(0, 0);
INSERT INTO CamereRezervate VALUES(1, 0);
INSERT INTO CamereRezervate VALUES(2, 0);
INSERT INTO CamereRezervate VALUES(6, 1);
INSERT INTO CamereRezervate VALUES(7, 1);
INSERT INTO CamereRezervate VALUES(8, 2);
INSERT INTO CamereRezervate VALUES(1, 3);
```

```

INSERT INTO CamereRezervate VALUES(2, 3);
INSERT INTO CamereRezervate VALUES(3, 3);
INSERT INTO CamereRezervate VALUES(4, 3);

```

The screenshot shows the Live SQL interface with a SQL Worksheet containing 21 INSERT statements. The statements are grouped into three sections: Hotel, Angajati, and Camera. The execution results show that each statement was successfully inserted.

```

1 INSERT INTO Hotel VALUES(0, 'Racova', 'Str. Stefan cel Mare, nr.10, Vaslui');
2 INSERT INTO Hotel VALUES(1, 'Europa', 'Str. Stefan cel Mare, nr.11, Vaslui');
3 INSERT INTO Hotel VALUES(2, 'Intercontinental', 'Bd. Elisabeta, Bucuresti');
4
5 INSERT INTO Angajati VALUES(0, 'Liviu', 'Bouruc', 123456, 'Director', NULL, 2);
6 INSERT INTO Angajati VALUES(1, 'Miruna', 'Vasilu', 123457, 'Manager', 0, 2);
7 INSERT INTO Angajati VALUES(2, 'Ciuni', 'Ciurescu', 123458, 'Bucatar-Se', 0, 2);
8 INSERT INTO Angajati VALUES(3, 'Mihai', 'Dobrescu', 123459, 'Receptioner', 1, 2);
9 INSERT INTO Angajati VALUES(4, 'David', 'Popa', 123410, 'Bucatar', 2, 2);
10 INSERT INTO Angajati VALUES(5, 'Andrei', 'Filip', 123353, 'CEO', NULL, 0);
11 INSERT INTO Angajati VALUES(6, 'Marius', 'Dum', 123334, 'Manager', NULL, 1);
12 INSERT INTO Angajati VALUES(7, 'Tode', 'Lavruc', 123355, 'Majordom', 6, 1);
13
14 INSERT INTO TipCamera VALUES(0, 'Camera Single', 1, 120);
15 INSERT INTO TipCamera VALUES(1, 'Apartment', 2, 200);
16 INSERT INTO TipCamera VALUES(2, 'Penthouse', 4, 500);
17
18 INSERT INTO Camera VALUES(0, 2, 0, 11);
19 INSERT INTO Camera VALUES(1, 2, 0, 12);
20 INSERT INTO Camera VALUES(2, 2, 1, 21);
21 INSERT INTO Camera VALUES(3, 2, 2, 22);

```

Execution results:

```

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

```

© 2021 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0

6 Cerința 6

La cerința 6 am făcut o procedură care afișează toți managerii unui hotel dat ca parametru. Aceștia sunt salvați într-o colecție de tip tabel indexat și afișați ulterior.

```

CREATE OR REPLACE PROCEDURE Ex6 (numeh Hotel.nume_hotel%TYPE)
AS
    TYPE tbl_idx IS TABLE OF Angajati%ROWTYPE INDEX BY PLS_INTEGER;
    t tbl_idx;
BEGIN
    SELECT * BULK COLLECT INTO t
    FROM Angajati
    WHERE desc_job = 'Manager';

    DBMS_OUTPUT.PUT_LINE('Sunt ' || t.COUNT || ' manageri la hotelul ' || numeh);
    FOR i IN t.FIRST..t.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(t(i).nume || ' ' || t(i).prenume);
    END LOOP;
END;
/
BEGIN
    Ex6('Intercontinental');
END;
/

```

Live SQL

Feedback
Help
bouruc.liviu2000@gmail.com

SQL Worksheet
Clear
Find
Actions
Save
Run

```

100 CREATE OR REPLACE PROCEDURE Ex6 (numeh Hotel.nume_hotel%TYPE)
101 AS
102 TYPE tbl_idx IS TABLE OF Angajati%ROWTYPE INDEX BY PLS_INTEGER;
103 t tbl_idx;
104 BEGIN
105 SELECT * BULK COLLECT INTO t
106 FROM Angajati
107 WHERE desc_job = 'Manager';
108 DBMS_OUTPUT.PUT_LINE('Sunt ' || t.COUNT || ' manageri la hotelul ' || numeh);
109 FOR i IN t.FIRST..t.LAST LOOP
110 DBMS_OUTPUT.PUT_LINE(t(i).nume || ' ' || t(i).prenume);
111 END LOOP;
112 END;
113 /
114 BEGIN
115 Ex6('Intercontinental');
116 END;
117 /

```

1 row(s) inserted.

1 row(s) inserted.

Procedure created.

Statement processed.

Sunt 2 manageri la hotelul Intercontinental

Mihnea Vasiliu

Marius Dumi

ORACLE
Integrated Cloud
Applications & Platform Services

© 2021 Oracle Corporation - Privacy - Terms of Use

Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter

Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0

Activate Windows

Go to Settings to activate Windows.

7 Cerința 7

La cerința 7 am făcut o procedură care afișează pentru fiecare hotel câți angajați lucrează acolo. Numele hotelurilor și numărul de angajați pentru fiecare sunt salvate într-un cursor.

CREATE OR REPLACE PROCEDURE Ex7

AS

v_nume Hotel.nume_hotel%TYPE;

v_nr NUMBER(4);

CURSOR c IS

SELECT nume_hotel, COUNT(id_angajat)

FROM Hotel h JOIN Angajati a ON (h.id_hotel = a.id_hotel)

GROUP BY nume_hotel;

BEGIN

OPEN c;

LOOP

FETCH c INTO v_nume, v_nr;

EXIT WHEN c%NOTFOUND;

IF v_nr = 0 THEN

DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_nume || ' nu lucreaza angajati');

ELSIF v_nr = 1 THEN

DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_nume || ' lucreaza un angajat');

ELSE

DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_nume || ' lucreaza ' || v_nr ||
↵ angajati');

END IF;

END LOOP;

CLOSE c;

END;

/

BEGIN

Ex7();

END;

/

```

1 CREATE OR REPLACE PROCEDURE Ex7
2 AS
3     v_name Hotel.nume_hotel%TYPE;
4     v_nr NUMBER(4);
5     CURSOR c IS
6         SELECT nume_hotel, COUNT(id_angajat)
7         FROM Hotel h JOIN Angajati a ON (h.id_hotel = a.id_hotel)
8         GROUP BY nume_hotel;
9 BEGIN
10     OPEN c;
11     LOOP
12         FETCH c INTO v_name, v_nr;
13         EXIT WHEN c%NOTFOUND;
14
15         IF v_nr = 0 THEN
16             DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_name || ' nu lucreaza angajati');
17         ELSEIF v_nr = 1 THEN
18             DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_name || ' lucreaza un angajat');
19         ELSE
20             DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_name || ' lucreaza ' || v_nr || ' angajati');
21         END IF;
22     END LOOP;
23     CLOSE c;
24 END;

```

Procedure created.

Statement processed.

In hotelul Intercontinental lucreaza 5 angajati
 In hotelul Racova lucreaza un angajat
 In hotelul Europa lucreaza 2 angajati

8 Cerința 8

La cerința 8 am făcut o funcție care returnează pentru un număr de cameră dat ca parametru câte rezervări au fost făcute la acea cameră, indiferent de hotel. Am identificat următoarele excepții care pot apărea:

1. niciun hotel nu are o cameră cu numărul dat
2. niciun client nu s-a cazat în camera respectivă
3. numărul camerei introdus este negativ
4. alte excepții care pot apărea

```

CREATE OR REPLACE FUNCTION Ex8(nrc NUMBER) RETURN NUMBER
IS
    nr_c_rez NUMBER;
    TYPE tbl_idx IS TABLE OF Camera%ROWTYPE INDEX BY PLS_INTEGER;
    aux tbl_idx;
    NEGATIVE_NUMBER EXCEPTION;
    NO_DATA_FOUND1 EXCEPTION;
    NO_DATA_FOUND2 EXCEPTION;
BEGIN
    IF nrc < 0 THEN
        RAISE NEGATIVE_NUMBER;
    END IF;
    SELECT * BULK COLLECT INTO aux FROM Camera WHERE nr_camera = nrc;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    SELECT COUNT(r.id_rezervare) INTO nr_c_rez
    FROM Rezervari r JOIN CamereRezervate cr ON (r.id_rezervare = cr.id_rezervare)
    JOIN Camera c ON (cr.id_camera = c.id_camera)
    WHERE c.nr_camera = nrc;

```



```

    IF nr_c_rez = 0 THEN
        RAISE NO_DATA_FOUND2;
    ELSE
        RETURN nr_c_rez;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista camera cu numarul ' || nrc);
        RETURN -1;
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista rezervari pentru camerele cu numarul ' || nrc);
        RETURN -1;
    WHEN NEGATIVE_NUMBER THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;
END;
/
DECLARE
    aux NUMBER;
BEGIN
    aux := Ex8(11);
    IF aux > -1 THEN
        DBMS_OUTPUT.PUT_LINE('Au fost inregistrate ' || aux || ' rezervari pentru
        ↪ camerele cu numarul 11');
    END IF;
END;
/

```

The screenshot shows the 'Live SQL' web interface. At the top, there's a navigation bar with 'Live SQL', 'Feedback', 'Help', and a user profile 'bouruc.liviu2000@gmail.com'. Below this is the 'SQL Worksheet' section with a toolbar containing 'Clear', 'Find', 'Actions', 'Save', and a 'Run' button. The main area displays a SQL script for creating a function 'Ex8' that takes a room number 'nrc' and returns the number of reservations. The script includes error handling for 'NO_DATA_FOUND1', 'NO_DATA_FOUND2', 'NEGATIVE_NUMBER', and 'OTHERS'. The function is then called with '11' in the 'BEGIN' block. The results pane on the right shows the output: 'Au fost inregistrate 2 rezervari pentru camerele cu numarul 11'.

```

1 CREATE OR REPLACE FUNCTION Ex8(nrc NUMBER) RETURN NUMBER
2 IS
3     nr_c_rez NUMBER;
4     NEGATIVE_NUMBER EXCEPTION;
5 BEGIN
6     IF nrc < 0 THEN
7         RAISE NEGATIVE_NUMBER;
8     END IF;
9
10    SELECT COUNT(r.id_rezervare) INTO nr_c_rez
11    FROM Rezervari r JOIN CamereRezervate cr ON (r.id_rezervare = cr.id_rezervare)
12    JOIN Camera c ON (cr.id_camera = c.id_camera)
13    WHERE c.nr_camera = nrc;
14
15    IF nr_c_rez = 0 THEN
16        RAISE NO_DATA_FOUND;
17    ELSE
18        RETURN nr_c_rez;
19    END IF;
20 EXCEPTION
21     WHEN NO_DATA_FOUND THEN

```

Function created.

Statement processed.

Au fost inregistrate 2 rezervari pentru camerele cu numarul 11

Live SQL

Feedback Help bouruc.liviu2000@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
21 WHERE c.nr_camera = nrc;
22
23 IF nr_c_rez = 0 THEN
24 RAISE NO_DATA_FOUND2;
25 ELSE
26 RETURN nr_c_rez;
27 END IF;
28 EXCEPTION
29 WHEN NO_DATA_FOUND1 THEN
30 DBMS_OUTPUT.PUT_LINE('Nu exista camera cu numarul ' || nrc);
31 RETURN -1;
32 WHEN NO_DATA_FOUND2 THEN
33 DBMS_OUTPUT.PUT_LINE('Nu exista rezervari pentru camerele cu numarul ' || nrc);
34 RETURN -1;
35 WHEN NEGATIVE_NUMBER THEN
36 DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
37 RETURN -1;
38 WHEN OTHERS THEN
39 DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
40 DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
41 RETURN -1;
```

Function created.

Statement processed.
Nu exista camera cu numarul 111

ORACLE Integrated Cloud
Applications & Platform Services

© 2021 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ❤️ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

Activate Windows
Go to Settings to activate Windows.

Live SQL

Feedback Help bouruc.liviu2000@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
16 DBMS_OUTPUT.PUT_LINE('Nu exista rezervari pentru camerele cu numarul ' || nrc);
17 ELSE
18 RETURN nr_c_rez;
19 END IF;
20 EXCEPTION
21 WHEN NO_DATA_FOUND THEN
22 DBMS_OUTPUT.PUT_LINE('Nu exista rezervari pentru camerele cu numarul ' || nrc);
23 RETURN -1;
24 WHEN NEGATIVE_NUMBER THEN
25 DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
26 RETURN -1;
27 WHEN OTHERS THEN
28 DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
29 DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
30 RETURN -1;
31 END;
32 /
33 DECLARE
34 aux NUMBER;
35 BEGIN
36 aux := Ex8(111);
```

Function created.

Statement processed.
Nu exista rezervari pentru camerele cu numarul 111

ORACLE Integrated Cloud
Applications & Platform Services

© 2021 Oracle Corporation - Privacy - Terms of Use
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter
Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ❤️ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

Activate Windows
Go to Settings to activate Windows.

```

17 ELSE
18     RETURN nr_c_rez;
19 END IF;
20 EXCEPTION
21 WHEN NO_DATA_FOUND THEN
22     DBMS_OUTPUT.PUT_LINE('Nu exista rezervari pentru camerele cu numarul ' || nrc);
23     RETURN -1;
24 WHEN NEGATIVE_NUMBER THEN
25     DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
26     RETURN -1;
27 WHEN OTHERS THEN
28     DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
29     DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
30     RETURN -1;
31 END;
32 /
33 DECLARE
34     aux NUMBER;
35 BEGIN
36     aux := Ex8(-1);

```

Function created.

Statement processed.

Nu sunt permise valori negative!

```

1 CREATE OR REPLACE FUNCTION Ex8(nrc NUMBER) RETURN NUMBER
2 IS
3     nr_c_rez NUMBER;
4     NEGATIVE_NUMBER EXCEPTION;
5     aux Camera%ROWTYPE;
6 BEGIN
7     IF nrc < 0 THEN
8         RAISE NEGATIVE_NUMBER;
9     END IF;
10    SELECT * INTO aux FROM Camera WHERE nr_camera = nrc;
11    IF SQL%NOTFOUND THEN
12        RAISE NO_DATA_FOUND;
13    END IF;
14
15    SELECT COUNT(r.id_rezervare) INTO nr_c_rez
16    FROM Rezervari r JOIN CamereRezervate cr ON (r.id_rezervare = cr.id_rezervare)
17    JOIN Camera c ON (cr.id_camera = c.id_camera)
18    WHERE c.nr_camera = nrc;
19
20    IF nr_c_rez = 0 THEN
21        RAISE NO_DATA_FOUND;

```

Function created.

Statement processed.

Codul erorii: -1422
 Mesajul erorii: ORA-01422: exact fetch returns more than requested number of rows

9 Cerința 9

La cerința 9 am făcut o procedură care afișează pentru un client dat ca parametru la ce hoteluri s-a cazat. Am identificat următoarele excepții care pot apărea:

1. nu există un client cu numele dat
2. clientul dat nu a fost cazat la niciun hotel
3. alte excepții care pot apărea

```

CREATE OR REPLACE PROCEDURE Ex9(nume VARCHAR2)
AS

```

```

    TYPE tbl_idx_h IS TABLE OF Hotel.nume_hotel%TYPE INDEX BY PLS_INTEGER;
    v_nume tbl_idx_h;
    TYPE tbl_idx_cl IS TABLE OF Clienti%ROWTYPE INDEX BY PLS_INTEGER;

```

```

    aux tbl_idx_cl;
    NO_DATA_FOUND1 EXCEPTION;
    NO_DATA_FOUND2 EXCEPTION;
BEGIN
    SELECT * BULK COLLECT INTO aux FROM Clienti WHERE nume_client = nume;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    SELECT nume_hotel BULK COLLECT INTO v_nume
    FROM Hotel h JOIN Camera c ON (h.id_hotel = c.id_hotel)
    JOIN CamereRezervate cr ON (c.id_camera = cr.id_camera)
    JOIN Rezervari r ON (cr.id_rezervare = r.id_rezervare)
    JOIN Clienti c ON (r.id_client = c.id_client)
    WHERE nume_client = nume;

    IF v_nume.COUNT = 0 THEN
        RAISE NO_DATA_FOUND2;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Hotelurile la care s-a cazat clientul cautat: ');
    FOR i IN v_nume.FIRST..v_nume.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(v_nume(i));
    END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clientul cu numele ' || nume);
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista date de afisat');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
END;
/
BEGIN
    Ex9('George');
END;
/

```

Live SQL

Feedback Help bounuc.liviu2000@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
1 CREATE OR REPLACE PROCEDURE Ex9(num VARCHAR2)
2 AS
3     TYPE tbl_idx_h IS TABLE OF Hotel.num_hotel%TYPE INDEX BY PLS_INTEGER;
4     v_num tbl_idx_h;
5     TYPE tbl_idx_cl IS TABLE OF Clienti%ROWTYPE INDEX BY PLS_INTEGER;
6     aux tbl_idx_cl;
7     NO_DATA_FOUND1 EXCEPTION;
8     NO_DATA_FOUND2 EXCEPTION;
9 BEGIN
10    SELECT * BULK COLLECT INTO aux FROM Clienti WHERE num_client = num;
11    IF SQL%NOTFOUND THEN
12        RAISE NO_DATA_FOUND1;
13    END IF;
14
15    SELECT num_hotel BULK COLLECT INTO v_num
16    FROM Hotel h JOIN Camera c ON (h.id_hotel = c.id_hotel)
17    JOIN CameraRezervare cr ON (c.id_camera = cr.id_camera)
18    JOIN Rezervari r ON (cr.id_rezervare = r.id_rezervare)
19    JOIN Clienti cl ON (r.id_client = c.id_client)
20    WHERE num_client = num;
21
```

Procedure created.

Statement processed.
Hotelurile la care s-a cazat clientul cautat:
Europa
Europa
Racova

Live SQL

Feedback Help bounuc.liviu2000@gmail.com

SQL Worksheet

Clear Find Actions Save Run

```
23 RAISE NO_DATA_FOUND2;
24 END IF;
25
26 DBMS_OUTPUT.PUT_LINE('Hotelurile la care s-a cazat clientul cautat: ');
27 FOR i IN v_num.FIRST..v_num.LAST LOOP
28     DBMS_OUTPUT.PUT_LINE(v_num(i));
29 END LOOP;
30 EXCEPTION
31 WHEN NO_DATA_FOUND1 THEN
32     DBMS_OUTPUT.PUT_LINE('Nu exista clientul cu numele ' || num);
33 WHEN NO_DATA_FOUND2 THEN
34     DBMS_OUTPUT.PUT_LINE('Nu exista date de afisat');
35 WHEN OTHERS THEN
36     DBMS_OUTPUT.PUT_LINE('Codul erorii: ' || SQLCODE);
37     DBMS_OUTPUT.PUT_LINE('Mesajul erorii: ' || SQLERRM);
38 END;
39 /
40 BEGIN
41     Ex9('Darius');
42 END;
43 /
```

Procedure created.

Statement processed.
Nu exista clientul cu numele Darius

Live SQL

Feedback

Help

bouruc.liviu2000@gmail.com

SQL Worksheet

Clear

Find

Actions

Save

Run

```

24      RAISE NO_DATA_FOUND;
25  END IF;
26
27  DBMS_OUTPUT.PUT_LINE('Hotelurile la care s-a cazat clientul cautat: ');
28  FOR i IN v_name.FIRST..v_name.LAST LOOP
29      DBMS_OUTPUT.PUT_LINE(v_name(i));
30  END LOOP;
31
32  EXCEPTION
33  WHEN NO_DATA_FOUND THEN
34      DBMS_OUTPUT.PUT_LINE('Nu exista clientul cu numele ' || nume);
35  WHEN NO_DATA_FOUND THEN
36      DBMS_OUTPUT.PUT_LINE('Nu exista date de afisat');
37  WHEN OTHERS THEN
38      DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
39      DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
40
41  END;
42  BEGIN
43      Ex9('Costea');
44  END;

```

1 row(s) Inserted.

Procedure created.

Statement processed.

Nu exista date de afisat

ORACLE

Integrated Cloud

Applications & Platform Services

© 2021 Oracle Corporation - Privacy - Terms of Use

Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter

Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0

Built with ❤️ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

Activate Windows

Go to Settings to activate Windows.

Live SQL

Feedback

Help

bouruc.liviu2000@gmail.com

SQL Worksheet

Clear

Find

Actions

Save

Run

```

1  CREATE OR REPLACE PROCEDURE Ex9(num VARCHAR2)
2  AS
3      TYPE tbl_idx_h IS TABLE OF Hotel.nume_hotel%TYPE INDEX BY PLS_INTEGER;
4      v_name tbl_idx_h;
5      TYPE tbl_idx_c1 IS TABLE OF Clienti%ROWTYPE INDEX BY PLS_INTEGER;
6      aux tbl_idx_c1;
7      aux2 Camera%ROWTYPE;
8      NO_DATA_FOUND1 EXCEPTION;
9      NO_DATA_FOUND2 EXCEPTION;
10 BEGIN
11     SELECT * INTO aux2 FROM Camera;
12     SELECT * BULK COLLECT INTO aux FROM Clienti WHERE nume_client = num;
13     IF SQL%NOTFOUND THEN
14         RAISE NO_DATA_FOUND1;
15     END IF;
16
17     SELECT nume_hotel BULK COLLECT INTO v_name
18     FROM Hotel h JOIN Camera c ON (h.id_hotel = c.id_hotel)
19     JOIN CamereRezervate cr ON (c.id_camera = cr.id_camera)
20     JOIN Rezervari r ON (cr.idRezervare = r.idRezervare)
21     JOIN Clienti c ON (r.id_client = c.id_client)

```

Procedure created.

Statement processed.

Codul erorii: -1422

Mesajul erorii: ORA-01422: exact fetch returns more than requested number of rows

ORACLE

Integrated Cloud

Applications & Platform Services

© 2021 Oracle Corporation - Privacy - Terms of Use

Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter

Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0

Built with ❤️ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

Activate Windows

Go to Settings to activate Windows.

10 Cerința 10

La cerința 10 am făcut un trigger care se declanșează atunci când încercăm să facem modificări asupra tabelului *Angajati* într-o zi nelucrătoare (25 decembrie sau 1 ianuarie).

```

CREATE OR REPLACE TRIGGER Ex10
BEFORE INSERT OR UPDATE OR DELETE ON Angajati
BEGIN
    IF (TO_CHAR(SYSDATE, 'DD/MM') = '25/12' OR TO_CHAR(SYSDATE, 'DD/MM') = '01/01' OR
        TO_CHAR(SYSDATE, 'DD/MM') = '06/01') THEN
        RAISE_APPLICATION_ERROR(-20001, 'In zilele de 25 decembrie si 1 ianuarie nu se
        lucreaza!');

```

14

```

        END IF;
    END;
/
--trebuie facuta inserarea in zilele mentionate mai sus; pentru demonstratie, am adaugat
↳ in if si ziua curenta
INSERT INTO Angajati VALUES(8, 'Demo', 'Demo', 1233554, 'Majordom', 6, 1);

```

The screenshot shows the Live SQL interface with the following SQL code in the editor:

```

1 CREATE OR REPLACE TRIGGER Ex10
2 BEFORE INSERT OR UPDATE OR DELETE ON Angajati
3 BEGIN
4     IF (TO_CHAR(SYSDATE, 'DD/MM') = '25/12' OR TO_CHAR(SYSDATE, 'DD/MM') = '01/01' OR TO_CHAR(SYSDATE, 'DD/MM') = '06/01') THEN
5         RAISE_APPLICATION_ERROR(-20001, 'In zilele de 25 decembrie si 1 ianuarie nu se lucreaza!');
6     END IF;
7 END;
8 /
9 --trebuie facuta inserarea in zilele mentionate mai sus; pentru demonstratie, am adaugat in if si ziua curenta
10 INSERT INTO Angajati VALUES(8, 'Demo', 'Demo', 1233554, 'Majordom', 6, 1);

```

The output pane shows the following messages:

```

Trigger created.
ORA-20001: In zilele de 25 decembrie si 1 ianuarie nu se lucreaza! ORA-06512: at "SQL_NLPCHQ4MCTFSKXMBR30FVG0.EX10", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

11 Cerința 11

La cerința 11 am făcut un trigger care se declanșează atunci când încercăm să facem modificări asupra coloanei *CNP* din tabelul *Angajati* (acesta fiind un câmp care nu ar trebui modificat niciodată).

```

CREATE OR REPLACE TRIGGER Ex11
    BEFORE UPDATE OF CNP ON Angajati
    FOR EACH ROW
    WHEN (NEW.CNP <> OLD.CNP)
BEGIN
    RAISE_APPLICATION_ERROR(-20002, 'Nu puteti modifica CNP-ul unei persoane');
END;
/
UPDATE Angajati SET CNP = 0;

```

Live SQL

Feedback
Help
bouruc.liviu2000@gmail.com

SQL Worksheet
Clear
Find
Actions
Save
Run

```

1 CREATE OR REPLACE TRIGGER Ex11
2 BEFORE UPDATE OF CNP ON Angajati
3 FOR EACH ROW
4 WHEN (NEW.CNP <> OLD.CNP)
5 BEGIN
6 RAISE_APPLICATION_ERROR(-20002, 'Nu puteti modifica CNP-ul unei persoane');
7 END;
8 /
9 UPDATE Angajati SET CNP = 0;

```

Trigger created.

ORA-20002: Nu puteti modifica CNP-ul unei persoane ORA-06512: at "SQL_ILPCVQAQKCTFSFKUHBKJOFVGO.EX11", line 2
ORA-06512: at "SYS.DBMS_SQL", line 1721

Integrated Cloud
Applications & Platform Services

© 2021 Oracle Corporation - Privacy - Terms of Use

Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c, 18c, 12c - Follow on Twitter

Live SQL 20.4.2, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0

Built with ❤️ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

Activate Windows
Go to Settings to activate Windows.

12:52

12 Cerința 12

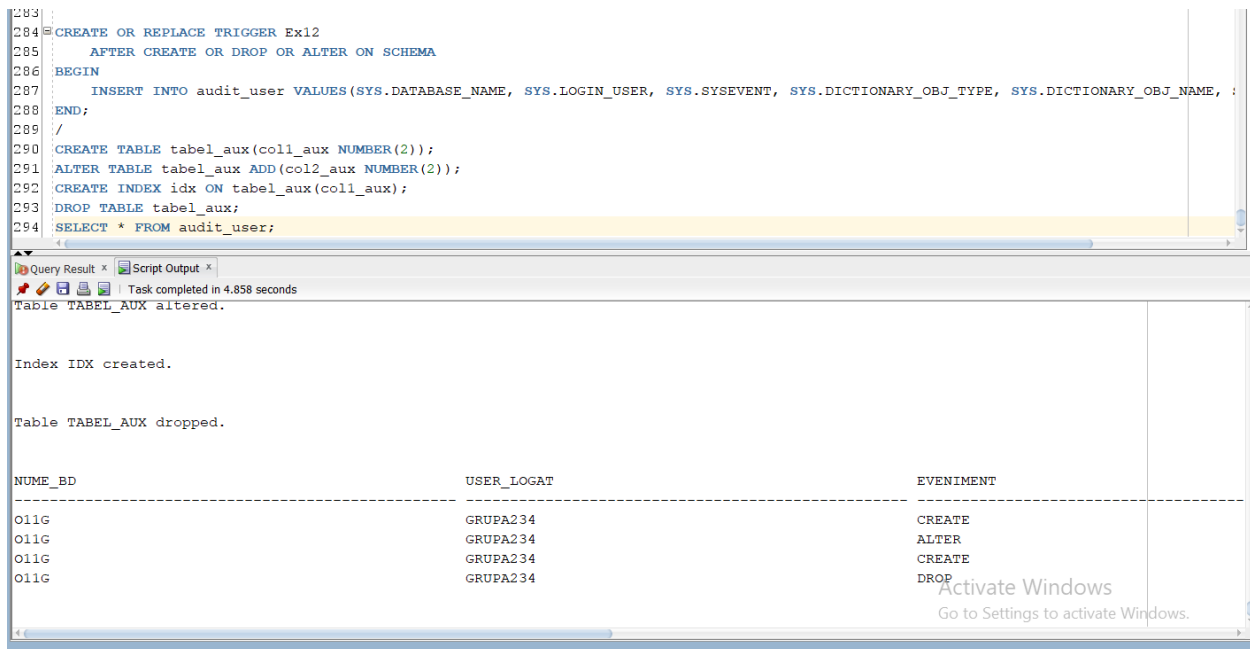
La cerința 12 am făcut un trigger care se declanșează de fiecare dată când se execută o operație de tip LDD, inserând în tabela nou creată un log cu detaliile operației.

```

CREATE TABLE audit_user(
    nume_bd VARCHAR2(50),
    user_logat VARCHAR2(50),
    eveniment VARCHAR2(50),
    tip_obiect_referit VARCHAR2(50),
    nume_obiect_referit VARCHAR2(50),
    data_exec TIMESTAMP(3));

CREATE OR REPLACE TRIGGER Ex12
AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO audit_user VALUES(SYS.DATABASE_NAME, SYS.LOGIN_USER, SYS.SYSEVENT,
    ↳ SYS.DICTIONARY_OBJ_TYPE, SYS.DICTIONARY_OBJ_NAME, SYSTIMESTAMP(3));
END;
/
CREATE TABLE tabel_aux(col1_aux NUMBER(2));
ALTER TABLE tabel_aux ADD(col2_aux NUMBER(2));
CREATE INDEX idx ON tabel_aux(col1_aux);
DROP TABLE tabel_aux;
SELECT * FROM audit_user;

```

13 Cerința 13

CREATE OR REPLACE PACKAGE Proiect

AS

PROCEDURE Ex6(numeh Hotel.nume_hotel%TYPE);

PROCEDURE Ex7;

FUNCTION Ex8(nrc NUMBER) RETURN NUMBER;

PROCEDURE Ex9(numa VARCHAR2);

END Proiect;

/

CREATE OR REPLACE PACKAGE BODY Proiect

AS

PROCEDURE Ex6 (numeh Hotel.nume_hotel%TYPE)

AS

TYPE tbl_idx IS TABLE OF Angajati%ROWTYPE INDEX BY PLS_INTEGER;

t tbl_idx;

BEGIN

SELECT * BULK COLLECT INTO t

FROM Angajati

WHERE desc_job = 'Manager';

DBMS_OUTPUT.PUT_LINE('Sunt ' || t.COUNT || ' manageri la hotelul ' || numeh);

FOR i IN t.FIRST..t.LAST LOOP

DBMS_OUTPUT.PUT_LINE(t(i).nume || ' ' || t(i).prenume);

END LOOP;

END;

PROCEDURE Ex7

AS

v_nume Hotel.nume_hotel%TYPE;

v_nr NUMBER(4);

CURSOR c IS

```

        SELECT nume_hotel, COUNT(id_angajat)
        FROM Hotel h JOIN Angajati a ON (h.id_hotel = a.id_hotel)
        GROUP BY nume_hotel;
BEGIN
    OPEN c;
    LOOP
        FETCH c INTO v_nume, v_nr;
        EXIT WHEN c%NOTFOUND;

        IF v_nr = 0 THEN
            DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_nume || ' nu lucreaza
            ↪ angajati');
        ELSIF v_nr = 1 THEN
            DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_nume || ' lucreaza un
            ↪ angajat');
        ELSE
            DBMS_OUTPUT.PUT_LINE('In hotelul ' || v_nume || ' lucreaza ' || v_nr
            ↪ || ' angajati');
        END IF;
    END LOOP;
    CLOSE c;
END;

FUNCTION Ex8(nrc NUMBER) RETURN NUMBER
IS
    nr_c_rez NUMBER;
    TYPE tbl_idx IS TABLE OF Camera%ROWTYPE INDEX BY PLS_INTEGER;
    aux tbl_idx;
    NEGATIVE_NUMBER EXCEPTION;
    NO_DATA_FOUND1 EXCEPTION;
    NO_DATA_FOUND2 EXCEPTION;
BEGIN
    IF nrc < 0 THEN
        RAISE NEGATIVE_NUMBER;
    END IF;
    SELECT * BULK COLLECT INTO aux FROM Camera WHERE nr_camera = nrc;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    SELECT COUNT(r.id_rezervare) INTO nr_c_rez
    FROM Rezervari r JOIN CamereRezervate cr ON (r.id_rezervare = cr.id_rezervare)
    JOIN Camera c ON (cr.id_camera = c.id_camera)
    WHERE c.nr_camera = nrc;

    IF nr_c_rez = 0 THEN
        RAISE NO_DATA_FOUND2;
    ELSE
        RETURN nr_c_rez;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista camera cu numarul ' || nrc);
        RETURN -1;

```

```

    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista rezervari pentru camerele cu numarul ' ||
            ↪ nrc);
        RETURN -1;
    WHEN NEGATIVE_NUMBER THEN
        DBMS_OUTPUT.PUT_LINE('Nu sunt permise valori negative!');
        RETURN -1;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
        RETURN -1;
END;

PROCEDURE Ex9(numa VARCHAR2)
AS
    TYPE tbl_idx_h IS TABLE OF Hotel.numa_hotel%TYPE INDEX BY PLS_INTEGER;
    v_numa tbl_idx_h;
    TYPE tbl_idx_cl IS TABLE OF Clienti%ROWTYPE INDEX BY PLS_INTEGER;
    aux tbl_idx_cl;
    NO_DATA_FOUND1 EXCEPTION;
    NO_DATA_FOUND2 EXCEPTION;
BEGIN
    SELECT * BULK COLLECT INTO aux FROM Clienti WHERE numa_client = numa;
    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND1;
    END IF;

    SELECT numa_hotel BULK COLLECT INTO v_numa
    FROM Hotel h JOIN Camera c ON (h.id_hotel = c.id_hotel)
    JOIN CamereRezervate cr ON (c.id_camera = cr.id_camera)
    JOIN Rezervari r ON (cr.id_rezervare = r.id_rezervare)
    JOIN Clienti c ON (r.id_client = c.id_client)
    WHERE numa_client = numa;

    IF v_numa.COUNT = 0 THEN
        RAISE NO_DATA_FOUND2;
    END IF;

    DBMS_OUTPUT.PUT_LINE('Hotelurile la care s-a cazat clientul cautat: ');
    FOR i IN v_numa.FIRST..v_numa.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(v_numa(i));
    END LOOP;
EXCEPTION
    WHEN NO_DATA_FOUND1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clientul cu numele ' || numa);
    WHEN NO_DATA_FOUND2 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista date de afisat');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE ('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE ('Mesajul erorii: ' || SQLERRM);
END;
END Proiect;
/

```

Live SQL

Feedback
Help
bouruc liviu2000@gmail.com

SQL Worksheet
Clear
Find
Actions
Save
Run

```

1 CREATE OR REPLACE PACKAGE Project
2 AS
3     PROCEDURE Ex6(numeh Hotel.num_hote1%TYPE);
4     PROCEDURE Ex7;
5     FUNCTION Ex8(nrc NUMBER) RETURN NUMBER;
6     PROCEDURE Ex9(num VARCHAR2);
7 END Project;
8 /
9 CREATE OR REPLACE PACKAGE BODY Project
10 AS
11     PROCEDURE Ex6 (numeh Hotel.num_hote1%TYPE)
12     AS
13         TYPE tbl_idx IS TABLE OF Angajat%ROWTYPE INDEX BY PLS_INTEGER;
14         t tbl_idx;
15     BEGIN
16         SELECT * BULK COLLECT INTO t
17         FROM Angajat1
18         WHERE desc_job = 'Manager';
19
20         DBMS_OUTPUT.PUT_LINE('Sunt ' || t.COUNT || ' manageri la hotelul ' || numeh);
21         FOR i IN t.FIRST .. t.LAST LOOP

```

Package created.

Package Body created.

© 2021 Oracle Corporation - Privacy - Terms of Use

Interacted Cloud
Oracle Learning Library - Ask Tom - Dev Gym - Database Doc 19c - 18c - 12c - Follow on Twitter

Activate Windows

14 Cerința 14

La cerința 14 am decis să fac o stivă, care în spate folosește un tablou imbricat.

```

CREATE OR REPLACE PACKAGE Stiva
AS
    PROCEDURE Adauga(x NUMBER);
    PROCEDURE Scoate;
    FUNCTION Top RETURN NUMBER;
    TYPE tbl_imb IS TABLE OF NUMBER;
    stiva tbl_imb;
END Stiva;
/
CREATE OR REPLACE PACKAGE BODY Stiva
AS
    PROCEDURE Adauga(x NUMBER) AS
    BEGIN
        stiva.EXTEND(x);
    END;

    PROCEDURE Scoate
    AS
        NO_ELEMENTS EXCEPTION;
    BEGIN
        IF stiva.COUNT > 0 THEN
            stiva.DELETE(stiva.LAST);
        ELSE
            RAISE NO_ELEMENTS;
        END IF;
    EXCEPTION
        WHEN NO_ELEMENTS THEN
            DBMS_OUTPUT.PUT_LINE('Stiva este goala');
    END;

    FUNCTION Top RETURN NUMBER AS

```

```

        BEGIN
            RETURN stiva.LAST;
        END;
    END Stiva;
/

```

Live SQL
Feedback
Help
bouruc.liviu2000@gmail.com

SQL Worksheet
Clear
Find
Actions
Save
Run

```

1 CREATE OR REPLACE PACKAGE Stiva
2 AS
3     PROCEDURE Aadauga(x NUMBER);
4     PROCEDURE Scoate;
5     FUNCTION Top RETURN NUMBER;
6     TYPE tbl_imp IS TABLE OF NUMBER;
7     stiva tbl_imp;
8 END Stiva;
9 /
10 CREATE OR REPLACE PACKAGE BODY Stiva
11 AS
12     PROCEDURE Aadauga(x NUMBER) AS
13     BEGIN
14         stiva.EXTEND(x);
15     END;
16
17     PROCEDURE Scoate
18     AS
19         NO_ELEMENTS EXCEPTION;
20     BEGIN
21         IF stiva.COUNT > 0 THEN

```

Package created.

Package Body created.