

The screenshot shows the SSMS interface with the following details:

- Database Explorer:** Shows the schema of the `project_sgbd` database, including tables like ANEXA, ANGAJAT, APROVIZIONARE, CAMERA, CLIENT, FACILITATE, and HOTEL.
- Code Editor:** Displays a PL/SQL trigger named `trigger_insert_hotel`:


```

13   info_hotel.info_hotel_ex10_tab := info_hotel_ex10_tab();
14
15   BEGIN
16     FOR v_info_hotel IN ( SELECT h.cod_hotel AS cod_hotel, COUNT(*) AS nr_rezervari, SUM(r.pret) AS incasari
17                           FROM rezervare r JOIN camera cam ON r.cod_camera = cam.cod_camera
18                           JOIN hotel h ON cam.cod_hotel = h.cod_hotel
19                           GROUP BY h.cod_hotel
20                           ORDER BY SUM(r.pret) DESC
21                           FETCH FIRST 2 ROWS ONLY
22                         )
23
24     LOOP
25       info_hotel.EXTEND;
26       info_hotel(info_hotel.LAST).nr_rezervari := v_info_hotel.nr_rezervari;
27       info_hotel(info_hotel.LAST).incasari := v_info_hotel.incasari;
28     END LOOP;
29     IF (info_hotel(1).incasari + info_hotel(2).incasari) / 2 < 21000 THEN
30       RAISE_APPLICATION_ERROR(-20000, 'Nu se poate deschide un nou hotel');
31     END IF;
32
33   END;
34
35   INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020', 'DD-MM-YYYY'), 44, 3);
36
37   INSERT INTO rezervare VALUES (19, 103, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 2000);
38   ROLLBACK;
      
```
- Output Window:** Shows the execution results of the trigger:


```

C##PROJECT_SGBD.HOTEL
    INTO_HOTEL(INTO_HOTEL.LAST).incasari := V_INTO_HOTEL.incasari;
    END LOOP;
    IF (info_hotel(1).incasari + info_hotel(2).incasari) / 2 < 21000 THEN
      RAISE_APPLICATION_ERROR(-20000, 'Nu se poate deschide un nou hotel');
    END IF;
      
```
- Services Tab:** Shows the transaction status (Tx) and database connection information.
- System Bar:** Includes icons for search, browser, file explorer, and other system utilities.

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

```

/* S-a stabilit ca bugetul pe care fiecare hotel il are pentru
aprovisionarea pentru anul 2023 sa nu depaseasca 11000 de lei.
Sa se verifice aceasta conditie la inserarea sau actualizarea unor
aprovisionari prin intermediul unui trigger. In cazul
in care operatia / operatiile fost facuta cu succes, sa se afiseze
costurile totale curente ale hotelurilor. */
      
```

Rezolvare:

-- varianta care da mutating table

```

CREATE OR REPLACE TRIGGER buget_aprovizionare_mutating
    BEFORE INSERT OR UPDATE ON aprovizionare
FOR EACH ROW
DECLARE
    TYPE aprovizionare_rec IS RECORD (
        cod_hotel aprovizionare.cod_hotel%TYPE,
      
```

```

        nume_hotel hotel.nume%TYPE,
        buget NUMBER);
TYPE aprovizionare_tab IS TABLE OF aprovizionare_rec INDEX BY
PLS_INTEGER;
costuri_aprovizionare aprovizionare_tab;

hotel_nou BOOLEAN := TRUE;

BEGIN
    SELECT ap.cod_hotel, h.nume, SUM(cost) BULK COLLECT INTO
costuri_aprovizionare
        FROM aprovizionare ap JOIN hotel h ON ap.cod_hotel = h.cod_hotel
        WHERE ap.data_aprovizionare BETWEEN TO_DATE('01-01-2023',
'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY')
        GROUP BY ap.cod_hotel, h.nume;

    hotel_nou := TRUE;
    -- daca data nu este din 2023, nu se face verificarea (conditia de
buget se aplica doar pe anul 2023)
    IF :NEW.data_aprovizionare BETWEEN TO_DATE('01-01-2023',
'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY') THEN
        FOR i IN 1..costuri_aprovizionare.COUNT LOOP
            IF costuri_aprovizionare(i).cod_hotel = :NEW.cod_hotel THEN
                IF INSERTING THEN
                    costuri_aprovizionare(i).buget :=
costuri_aprovizionare(i).buget + :NEW.cost;
                ELSIF UPDATING THEN
                    costuri_aprovizionare(i).buget :=
costuri_aprovizionare(i).buget - :OLD.cost + :NEW.cost;
                END IF;
                IF costuri_aprovizionare(i).buget > 11000 THEN
                    RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost
depasit');
                END IF;
                hotel_nou := FALSE;
                EXIT;
            END IF;
        END LOOP;
        IF hotel_nou THEN
            costuri_aprovizionare(costuri_aprovizionare.COUNT +
1).cod_hotel := :NEW.cod_hotel;
            SELECT nume INTO

```

```

costuri_aprovizionare(costuri_aprovizionare.COUNT).nume_hotel
    FROM hotel
    WHERE cod_hotel = :NEW.cod_hotel;
    costuri_aprovizionare(costuri_aprovizionare.COUNT).buget := :NEW.cost;
    IF costuri_aprovizionare(costuri_aprovizionare.COUNT).buget > 11000 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost
deposit');
        END IF;
    END IF;
END IF;

DBMS_OUTPUT.PUT_LINE('Costurile totale ale hotelurilor pentru anul
2023 sunt: ');
FOR i IN 1..costuri_aprovizionare.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE('Costul total al hotelului ' ||
costuri_aprovizionare(i).nume_hotel || ' este ' ||
costuri_aprovizionare(i).buget);
    END LOOP;
END;
/

UPDATE aprovizionare
SET cost = 1000
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare =
TO_DATE('23-04-2023', 'DD-MM-YYYY');

ALTER TRIGGER buget_aprovizionare_mutating DISABLE;

-- rezolvare mutating table: trigger compus

CREATE OR REPLACE TRIGGER buget_aprovizionare
    FOR INSERT OR UPDATE ON aprovizionare
COMPOUND TRIGGER

    TYPE aprovizionare_rec IS RECORD (
        cod_hotel aprovizionare.cod_hotel%TYPE,
        nume_hotel hotel.nume%TYPE,
        buget NUMBER);

```

```

TYPE aprovisionare_tab IS TABLE OF aprovisionare_rec INDEX BY
PLS_INTEGER;
costuri_aprovizionare aprovisionare_tab;

hotel_nou BOOLEAN := TRUE;

BEFORE STATEMENT IS
BEGIN
    SELECT ap.cod_hotel, h.nume, SUM(cost) BULK COLLECT INTO
costuri_aprovizionare
        FROM aprovisionare ap JOIN hotel h ON ap.cod_hotel = h.cod_hotel
        WHERE ap.data_aprovizionare BETWEEN TO_DATE('01-01-2023',
'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY')
        GROUP BY ap.cod_hotel, h.nume;
END BEFORE STATEMENT;

BEFORE EACH ROW IS
BEGIN
    hotel_nou := TRUE;
    -- daca data nu este din 2023, nu se face verificarea (conditia de
buget se aplica doar pe anul 2023)
    IF :NEW.data_aprovizionare BETWEEN TO_DATE('01-01-2023',
'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY') THEN
        FOR i IN 1..costuri_aprovizionare.COUNT LOOP
            IF costuri_aprovizionare(i).cod_hotel = :NEW.cod_hotel THEN
                IF INSERTING THEN
                    costuri_aprovizionare(i).buget :=
costuri_aprovizionare(i).buget + :NEW.cost;
                ELSIF UPDATING THEN
                    costuri_aprovizionare(i).buget :=
costuri_aprovizionare(i).buget - :OLD.cost + :NEW.cost;
                END IF;
                IF costuri_aprovizionare(i).buget > 11000 THEN
                    RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost
depasit');
                END IF;
                hotel_nou := FALSE;
                EXIT;
            END IF;
        END LOOP;
        IF hotel_nou THEN
            costuri_aprovizionare(costuri_aprovizionare.COUNT +
1).cod_hotel := :NEW.cod_hotel;
        END IF;
    END IF;
END;

```

```

        SELECT nume INTO
costuri_aprovizionare(costuri_aprovizionare.COUNT).nume_hotel
        FROM hotel
        WHERE cod_hotel = :NEW.cod_hotel;
        costuri_aprovizionare(costuri_aprovizionare.COUNT).buget := :NEW.cost;
                IF costuri_aprovizionare(costuri_aprovizionare.COUNT).buget > 11000 THEN
                        RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost depasit');
                END IF;
        END IF;
        END IF;
    END BEFORE EACH ROW;

AFTER STATEMENT IS
BEGIN
        DBMS_OUTPUT.PUT_LINE('Costurile totale ale hotelurilor pentru anul 2023 sunt: ');
        FOR i IN 1..costuri_aprovizionare.COUNT LOOP
                DBMS_OUTPUT.PUT_LINE('Costul total al hotelului ' ||
costuri_aprovizionare(i).nume_hotel || ' este ' ||
costuri_aprovizionare(i).buget);
        END LOOP;
        END AFTER STATEMENT;

END;
/

INSERT INTO aprovizionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 2050);
INSERT INTO aprovizionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 150);
UPDATE aprovizionare
SET cost = 10000
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
UPDATE aprovizionare
SET cost = 99
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');

ROLLBACK;

```

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\Scratches and Consoles\Database Consoles\project_sgbd\console_5 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_1 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_2 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_3 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_4 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_5 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_6 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_8 [project_sgbd]
- vlad\console [vlad]
- vlad\console_1 [vlad]
- vlad\console_2 [vlad]
- vlad\console_3 [vlad]
- vlad\console_4 [vlad]

console_5 [project_sgbd] x console_4 [vlad]

```

55
56
57     DBMS_OUTPUT.PUT_LINE( A ) 'Costurile totale ale hotelurilor pentru anul 2023 sunt: ';
58
59     FOR i IN 1 .. costuri_aprovizionare.COUNT LOOP
60         DBMS_OUTPUT.PUT_LINE( A || 'Costul total al hotelului ' || costuri_aprovizionare(i).nume_hotel || ' este ' || costuri_aprovizionare(i).valoare );
61     END LOOP;
62
63
64 UPDATE aprovisionare
65 SET cost = 10000
66 WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
67
68 ALTER TRIGGER buget_aprovizionare_mutating DISABLE;
69
70

```

[42000][4091]
ORA-04091: table C##PROJECT_SGBD.APROVIZIONARE is mutating, trigger/function may not see it
ORA-06512: at "C##PROJECT_SGBD.BUGET_APROVIZIONARE_MUTATING", line 12
ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.BUGET_APROVIZIONARE_MUTATING'
Position: 7

Services

Tx	Output	C##PROJECT_SGBD.APROVIZIONARE
✗	console_3 53 ms	(2024-01-08 23:41:31) ORA-04091: table C##PROJECT_SGBD.APROVIZIONARE is mutating, trigger/function may not see it
✗	console_5 282 ms	(2024-01-08 23:41:31) ORA-06512: at "C##PROJECT_SGBD.BUGET_APROVIZIONARE_MUTATING", line 12
✗	console_5 282 ms	(2024-01-08 23:41:31) ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.BUGET_APROVIZIONARE_MUTATING'
✗	vlad	
✗	console_4 42 s 391 ms	(2024-01-08 23:41:31) Position: 7

Database Consoles > project_sgbd > console_5 [project_sgbd]

64:8 CRLF UTF-8 4 spaces ENG 11:41 PM 1/8/2024

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\Scratches and Consoles\Database Consoles\project_sgbd\console_5 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_1 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_2 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_3 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_4 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_5 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_6 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project_sgbd\console_8 [project_sgbd]
- vlad\console [vlad]
- vlad\console_1 [vlad]
- vlad\console_2 [vlad]
- vlad\console_3 [vlad]
- vlad\console_4 [vlad]

console_5 [project_sgbd] x console_4 [vlad]

```

135
136
137     END;
138     /
139
140     INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 2050);
141     INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 150);
142     UPDATE aprovisionare
143     SET cost = 10000
144     WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
145     UPDATE aprovisionare
146     SET cost = 99
147     WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
148
149     ROLLBACK;
150     select * from APROVIZIONARE;
151     COMMIT;
152

```

[72000][20001]
ORA-20001: Bugetul a fost depasit
ORA-06512: at "C##PROJECT_SGBD.BUGET_APROVIZIONARE", line 33
ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.BUGET_APROVIZIONARE'
Position: 12

Services

Tx	Output	C##PROJECT_SGBD.APROVIZIONARE
✓	console_3 53 ms	(2024-01-08 23:42:27) ORA-20001: Bugetul a fost depasit
✗	console_5 31 ms	(2024-01-08 23:42:27) ORA-06512: at "C##PROJECT_SGBD.BUGET_APROVIZIONARE", line 33
✗	vlad	
✗	console_4 42 s 391 ms	(2024-01-08 23:42:27) ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.BUGET_APROVIZIONARE'

Database Consoles > project_sgbd > console_5 [project_sgbd]

138:13 CRLF UTF-8 4 spaces ENG 11:42 PM 1/8/2024

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\Scratches and Consoles\Database Consoles\project\project_sgbd\console [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_1 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_2 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_3 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_4 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_5 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_6 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_8 [project_sgbd]
- vlad\console [vlad]
- vlad\console_1 [vlad]
- vlad\console_2 [vlad]
- vlad\console_3 [vlad]

Tx: Manual

```

135 END;
136 /
137
138 INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 2050);
139 ✓
140 INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 150);
141 UPDATE aprovisionare
142 SET cost = 10000
143 WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
144 UPDATE aprovisionare
145 SET cost = 99
146 WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
147
148 ROLLBACK;
149 select * from APROVIZIONARE;
COMMIT;

```

Services

Tx

- console_1
- console_3 53 ms
- console_5 41 ms
- vlad
- console_4 42 s 391 ms
- console_4 42 s

Output C##PROJECT_SGBD.APROVIZIONARE

```

[2024-01-08 23:42:39] 1 row affected in 4 ms
Costurile totale ale hotelurilor pentru anul 2023 sunt:
Costul total al hotelului Novotel este 9500
Costul total al hotelului Ramada Plaza este 4600
Costul total al hotelului Riviera este 4800
Costul total al hotelului Alpin este 6400
Costul total al hotelului Vega este 10650

```

Database Consoles > project_sgbd > console_5 [project_sgbd]

139:3 CRLF UTF-8 4 spaces

Cloudy 11:42 PM 1/8/2024

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\Scratches and Consoles\Database Consoles\project\project_sgbd\console [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_1 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_2 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_3 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_4 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_5 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_6 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_8 [project_sgbd]
- vlad\console [vlad]
- vlad\console_1 [vlad]
- vlad\console_2 [vlad]
- vlad\console_3 [vlad]

Tx: Manual

```

135 END;
136 /
137
138 INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 2050);
139 ✓
140 INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 150);
141 UPDATE aprovisionare
142 SET cost = 10000
143 WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
144 UPDATE aprovisionare
145 SET cost = 99
146 WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
147
148 ROLLBACK;

```

Services

Tx

- console_1
- console_3 53 ms
- console_5 35 ms
- vlad
- console_4 42 s 391 ms
- console_4 42 s

Output C##PROJECT_SGBD.APROVIZIONARE

```

[72000][20001]
ORA-20001: Bugetul a fost depasit
ORA-06512: at "C##PROJECT_SGBD.BUGET_APROVIZIONARE", line 33
ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.BUGET_APROVIZIONARE'
Position: 7

```

Database Consoles > project_sgbd > console_5 [project_sgbd]

140:8 CRLF UTF-8 4 spaces

Cloudy 11:43 PM 1/8/2024

The screenshot shows the DataGrip IDE interface with the following details:

- File Explorer:** Shows a tree structure of database projects and consoles. The current project is "sqlproject" and the active console is "console_5 [project_sgbdb]".
- Code Editor:** Displays T-SQL code for provisioning data into a table. The code includes an `END;` statement, two `INSERT INTO` statements, an `UPDATE` statement, and a `ROLLBACK` statement.
- Output Window:** Shows the results of the executed queries, including the total cost for each hotel and the total cost for each hotel chain.
- Services Tab:** Shows the transaction history with details like query ID, duration, and execution time.

```
END;
/
INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 2050);
INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 150);
UPDATE aprovisionare
SET cost = 10000
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovisionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
UPDATE aprovisionare
SET cost = 99
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovisionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');

ROLLBACK;
select * from APROVIZIONARE;
COMMIT;
```

TX	ID	Console	Duration	Time	Message
✓	1	console_1	0 ms		
✗	2	console_3	53 ms	[2024-01-08 25:45:14]	1 row affected in 10 ms
✗	3	console_3	53 ms		Costurile totale ale hotelurilor pentru anul 2023 sunt:
✗	4	console_5	33 ms		Costul total al hotelului Novotel este 9500
✗	5	console_5	33 ms		Costul total al hotelului Ramada Plaza este 4600
✗	6	vlad			Costul total al hotelului Riviera este 4800
✗	7	console_4	42 s 391 ms		Costul total al hotelului Alpin este 6400
✗	8	console_4	42 s		Costul total al hotelului Vega este 9599

```
1  /* S-a stabilit ca bugetul pe care fiecare hotel il are pentru aprovizionarea pentru anul 2023 sa nu depaseasca 11000 de lei.  
2  Sa se verifice aceasta conditie la inserarea sau actualizarea unor aprovizionari prin intermediul unui trigger. In cazul  
3  in care operatia / operatiile sunt facute cu succes, sa se afiseze costurile totale curente ale hotelurilor. */  
4  
5  -- varianta care da mutating table  
6  
7  CREATE OR REPLACE TRIGGER buget_aprovizionare_mutating  
8      BEFORE INSERT OR UPDATE ON aprovizionare  
9      FOR EACH ROW  
10     DECLARE  
11         TYPE aprovizionare_rec IS RECORD (  
12             cod_hotel aprovizionare.cod_hotel%TYPE,  
13             nume_hotel hotel.nume%TYPE,  
14             buget NUMBER);  
15         TYPE aprovizionare_tab IS TABLE OF aprovizionare_rec INDEX BY PLS_INTEGER;  
16         costuri_aprovizionare aprovizionare_tab;  
17  
18         hotel_nou BOOLEAN := TRUE;  
19  
20     BEGIN  
21         SELECT ap.cod_hotel, h.nume, SUM(cost) BULK COLLECT INTO costuri_aprovizionare  
22             FROM aprovizionare ap JOIN hotel h ON ap.cod_hotel = h.cod_hotel  
23             WHERE ap.data_aprovizionare BETWEEN TO_DATE('01-01-2023', 'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY')  
24             GROUP BY ap.cod_hotel, h.nume;  
25  
26  
27         hotel_nou := TRUE;  
28         -- daca data nu este din 2023, nu se face verificarea (conditia de buget se aplică doar pe anul 2023)  
29         IF :NEW.data_aprovizionare BETWEEN TO_DATE('01-01-2023', 'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY') THEN  
30             FOR i IN 1..costuri_aprovizionare.COUNT LOOP  
31                 IF costuri_aprovizionare(i).cod_hotel = :NEW.cod_hotel THEN
```

sqlproject Version control

Files ▾

- sqlproject C:\Users\vcupuD
- Scratches and Consoles
 - Database Consoles
 - project
 - console [project_sgbd]
 - console_1 [project_sgbd]
 - console_2 [project_sgbd]
 - console_3 [project_sgbd]
 - console_4 [project_sgbd]
 - console_5 [project_sgbd]
 - console_6 [project_sgbd]
 - console_7 [project_sgbd]
 - console_8 [project_sgbd]
 - vlad
 - console [vlad]
 - console_1 [vlad]
 - console_2 [vlad]
 - console_3 [vlad]
 - console_4 [vlad]
- Extensions

Tx: Manual ▾

Playground ▾

IF costuri_aprovizionare(i).cod_hotel = :NEW.cod_hotel THEN
 IF INSERTING THEN
 costuri_aprovizionare(i).buget := costuri_aprovizionare(i).buget + :NEW.cost;
 ELSIF UPDATING THEN
 costuri_aprovizionare(i).buget := costuri_aprovizionare(i).buget - :OLD.cost + :NEW.cost;
 END IF;
 IF costuri_aprovizionare(i).buget > 11000 THEN
 RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost depasit');
 END IF;
 hotel_nou := FALSE;
 EXIT;
END IF;
END LOOP;
IF hotel_nou THEN
 costuri_aprovizionare(costuri_aprovizionare.COUNT + 1).cod_hotel := :NEW.cod_hotel;
 SELECT nume INTO costuri_aprovizionare(costuri_aprovizionare.COUNT).nume_hotel
 FROM hotel
 WHERE cod_hotel = :NEW.cod_hotel;
 costuri_aprovizionare(costuri_aprovizionare.COUNT).buget := :NEW.cost;
 IF costuri_aprovizionare(costuri_aprovizionare.COUNT).buget > 11000 THEN
 RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost depasit');
 END IF;
 END IF;
END IF;

DBMS_OUTPUT.PUT_LINE(A || 'Costurile totale ale hotelurilor pentru anul 2023 sunt: ');
FOR i IN 1 .. costuri_aprovizionare.COUNT LOOP
 DBMS_OUTPUT.PUT_LINE(A || 'Costul total al hotelului ' || costuri_aprovizionare(i).nume_hotel || ' este ' || costuri_aprovizionare(i).buget);
END LOOP;
END;

72:29 CRLF UTF-8 4 spaces ⌂ 11:44 PM 1/8/2024

sqlproject Version control

Files ▾

- sqlproject C:\Users\vcupuD
- Scratches and Consoles
 - Database Consoles
 - project
 - console [project_sgbd]
 - console_1 [project_sgbd]
 - console_2 [project_sgbd]
 - console_3 [project_sgbd]
 - console_4 [project_sgbd]
 - console_5 [project_sgbd]
 - console_6 [project_sgbd]
 - console_7 [project_sgbd]
 - console_8 [project_sgbd]
 - vlad
 - console [vlad]
 - console_1 [vlad]
 - console_2 [vlad]
 - console_3 [vlad]
 - console_4 [vlad]
- Extensions

Tx: Manual ▾

Playground ▾

UPDATE aprovisionare
SET cost = 10000
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
ALTER TRIGGER buget_aprovizionare_mutating DISABLE;
-- rezolvare mutating table: trigger compus
CREATE OR REPLACE TRIGGER buget_aprovizionare
 FOR INSERT OR UPDATE ON aprovisionare
 COMPOUND TRIGGER
 TYPE aprovisionare_rec IS RECORD (
 cod_hotel aprovisionare.cod_hotel%TYPE,
 nume_hotel hotel.nume%TYPE,
 buget NUMBER);
 TYPE aprovisionare_tab IS TABLE OF aprovisionare_rec INDEX BY PLS_INTEGER;
 costuri_aprovizionare aprovisionare_ap;
 hotel_nou BOOLEAN := TRUE;
 BEFORE STATEMENT IS
 BEGIN
 SELECT ap.cod_hotel, h.nume, SUM(cost) BULK COLLECT INTO costuri_aprovizionare
 FROM aprovisionare ap JOIN hotel h ON ap.cod_hotel = h.cod_hotel
 WHERE ap.data_aprovizionare BETWEEN TO_DATE('01-01-2023', 'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY')
 GROUP BY ap.cod_hotel, h.nume;
 END BEFORE STATEMENT;

72:29 CRLF UTF-8 4 spaces ⌂ 11:44 PM 1/8/2024

```
BEFORE EACH ROW IS
BEGIN
    hotel_nou := TRUE;
    -- daca data nu este din 2023, nu se face verificarea (conditia de buget se aplica doar pe anul 2023)
    IF :NEW.data_aprovizionare BETWEEN TO_DATE('01-01-2023', 'DD-MM-YYYY') AND TO_DATE('31-12-2023', 'DD-MM-YYYY') THEN
        FOR i IN 1..costuri_aprovizionare.COUNT LOOP
            IF costuri_aprovizionare(i).cod_hotel = :NEW.cod_hotel THEN
                IF INSERTING THEN
                    costuri_aprovizionare(i).buget := costuri_aprovizionare(i).buget + :NEW.cost;
                ELSIF UPDATING THEN
                    costuri_aprovizionare(i).buget := costuri_aprovizionare(i).buget - :OLD.cost + :NEW.cost;
                END IF;
                IF costuri_aprovizionare(i).buget > 11000 THEN
                    RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost depasit');
                END IF;
                hotel_nou := FALSE;
                EXIT;
            END IF;
        END LOOP;
        IF hotel_nou THEN
            costuri_aprovizionare(costuri_aprovizionare.COUNT + 1).cod_hotel := :NEW.cod_hotel;
            SELECT nume INTO costuri_aprovizionare(costuri_aprovizionare.COUNT).nume_hotel
            FROM hotel
            WHERE cod_hotel = :NEW.cod_hotel;
            costuri_aprovizionare(costuri_aprovizionare.COUNT).buget := :NEW.cost;
            IF costuri_aprovizionare(costuri_aprovizionare.COUNT).buget > 11000 THEN
                RAISE_APPLICATION_ERROR(-20001, 'Bugetul a fost depasit');
            END IF;
        END IF;
    END IF;
END BEFORE EACH ROW;
```

```
END IF;
END IF;
END IF;
END BEFORE EACH ROW;

AFTER STATEMENT IS
BEGIN
    DBMS_OUTPUT.PUT_LINE( A: 'Costurile totale ale hotelurilor pentru anul 2023 sunt: ');
    FOR i IN 1..costuri_aprovizionare.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE( A: 'Costul total al hotelului ' || costuri_aprovizionare(i).nume_hotel || ' este ' || costuri_aprovizionare(i).cost );
    END LOOP;
END AFTER STATEMENT;

END;
/

INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 2050);
INSERT INTO aprovisionare VALUES (11, 10, TO_DATE('09-01-2023', 'DD-MM-YYYY'), 150);
UPDATE aprovisionare
SET cost = 10000
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');
UPDATE aprovisionare
SET cost = 99
WHERE cod_partener = 11 AND cod_hotel = 10 AND data_aprovizionare = TO_DATE('23-04-2023', 'DD-MM-YYYY');

ROLLBACK;
select * from APROVIZIONARE;
```

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

```
/* Sa se inregistreze intr-un tabel toate erorile aparute ce apartin schemei, folosind un tabel auxiliar in care sa se retina numele bazei de date, userul care a generat eroarea, data la care aceasta a fost facuta, codul si numele erorii. */
```

Rezolvare:

```
CREATE SEQUENCE seq_id_eroare START WITH 1 INCREMENT BY 1;

CREATE TABLE erori(
    id_eroare NUMBER PRIMARY KEY,
    baza_de_date VARCHAR2(50),
    user_eroare VARCHAR2(50),
    data TIMESTAMP,
    err_stack VARCHAR2(2000)
);

CREATE OR REPLACE TRIGGER trig_eroare
AFTER SERVERERROR ON SCHEMA
BEGIN
    INSERT INTO erori VALUES(seq_id_eroare.NEXTVAL,  SYS.DATABASE_NAME,
    SYS.LOGIN_USER, SYSTIMESTAMP, DBMSUTILITY FORMAT_ERROR_STACK);
END;
/

CREATE TABLE tab_aux_eroare (id NUMBER(2) PRIMARY KEY);
INSERT INTO tab_aux_eroare VALUES (123);
INSERT INTO tab_aux_eroare VALUES (12);
INSERT INTO tab_aux_eroare VALUES (12);
SELECT * FROM tab_eroare_aux;
drop table tab_aux_eroare;

SELECT * FROM erori;
```

Database Explorer

Object Explorer

Services

Output

ID_ERORE	BAZA_DE_DATE	USER_ERORE	DATA	ERR_STACK
1	1	XE	C##PROJECT_SGBD	2024-01-05 02:53:29.771000 ORA-01438: value larger than specified precision allowed.
2	2	XE	C##PROJECT_SGBD	2024-01-05 02:53:42.829000 ORA-00001: unique constraint (C##PROJECT_SGBD.SYS_C0...).
3	3	XE	C##PROJECT_SGBD	2024-01-05 02:53:44.844000 ORA-00942: table or view does not exist.

Console 6 [project_sgbdb] > console_6 [project_sgbdb]

The screenshot shows the Oracle SQL Developer interface with the following details:

- Database Explorer** pane:
 - Project: project_2 of 40
 - Schema: project_sgbd 2 of 41
 - Tables (14): ANEXA, ANGAJAT, APROVIZIONARE, CAMERA, CLIENT, ERORI, FACILITATE, HOTEL.
 - Columns (6): COD_HOTEL, COD_LOCAT, NUME, DATA_INFIIN, NR_CAMERE, STELE.
 - Keys (1).
- Code Editor** pane (C##PROJECT_SGBD):

```
1 /* Sa se inregistreze intr-un tabel toate erorile aparute ce aparțin schemei, folosind un tabel auxiliar
2 în care să se retină numele bazei de date, userul care a generat eroarea, data la care aceasta a fost făcută,
3 codul și numele erorii. */
4
5 CREATE SEQUENCE seq_id_eroare START WITH 1 INCREMENT BY 1;
6
7 CREATE TABLE eroari(
8     id_eroare NUMBER PRIMARY KEY,
9     baza_de_date VARCHAR2(50),
10    user_eroare VARCHAR2(50),
11    data TIMESTAMP,
12    err_stack VARCHAR2(2000)
13 );
14
15 CREATE OR REPLACE TRIGGER trig_eroare
16 AFTER SERVERERROR ON SCHEMA
17 BEGIN
18     INSERT INTO eroari VALUES(seq_id_eroare.NEXTVAL, SYS.DBNAME, SYS.LOGINUSER, SYSTIMESTAMP, DBMSUTILITYFORMAT_ERROR_STACK);
19 END;
20 /
```
- Services** pane: Shows a list of services including console_5, console_6, and Vlad.
- Output** pane (C##PROJECT_SGBD.ERORI):

ID_EROARE	BAZA_DE_DATE	USER_EROARE	DATA	ERR_STACK
1	XE	C##PROJECT_SGBD	2024-01-05 02:53:42.771000	ORA-01438: value larger than specified precision allowed.
2	XE	C##PROJECT_SGBD	2024-01-05 02:53:42.829000	ORA-00001: unique constraint (C##PROJECT_SGBD.SYS_C0...
3	XE	C##PROJECT_SGBD	2024-01-05 02:53:44.844000	ORA-00942: table or view does not exist#
- Database Consoles**: project_sgbd > console_6 [project_sgbd]
- Bottom Bar**: Includes icons for file operations, search, navigation, and system status (7°C, mostly cloudy).

The screenshot shows the Oracle SQL Developer interface. The Database Explorer pane on the left lists tables such as ANEXA, ANGAJAT, APROVIZIONARE, and others under the schema C##PROJECT_SGBD. The central pane displays a PL/SQL script for creating a trigger named 'trig_eroare' that inserts into a temporary table 'erori1' whenever an error occurs. The Services tab shows a table named 'ERORI' with three rows of data. The bottom status bar indicates it's 2:54 AM on January 5, 2024.

13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

Rezolvare:

```

CREATE OR REPLACE PACKAGE hoteluri_pachet_ex13 IS
    PROCEDURE afisare_rezervari_hotel_ex6(nr_rezervari NUMBER, nume_hotel
    VARCHAR2);

    PROCEDURE rezervari_categorie_pret(categ_rezervare VARCHAR2);

    TYPE lista_nume_ex8 IS VARRAY(30) OF VARCHAR2(30);
    FUNCTION lista_ang_ex8(data_param hotel.data_infiintare%TYPE,
    partener_param partener.nume%TYPE) RETURN lista_nume_ex8;

    PROCEDURE afisare_clienti_hoteluri_piscina(cost_total_param IN NUMBER,
    nume_client_param OUT VARCHAR2);
END hoteluri_pachet_ex13;

```

```

CREATE OR REPLACE PACKAGE BODY hoteluri_pachet_ex13 IS
    PROCEDURE afisare_rezervari_hotel_ex6(nr_rezervari NUMBER, nume_hotel
VARCHAR2) IS
        -- varray pentru lista datelor de sosire
        TYPE lista_date IS VARRAY(20) OF rezervare.data_sosire%TYPE;
        TYPE info_client_ex6 IS RECORD(
            cod_client client.cod_client%TYPE,
            nume_client client.nume%TYPE,
            lista_date_sosire lista_date
        );
        -- tablou imbricat pentru a retine informatiile despre clienti
        TYPE lista_info_clienti_ex6 IS TABLE OF info_client_ex6;
        lista_info_clienti lista_info_clienti_ex6 :=
        lista_info_clienti_ex6();
        -- tablou indexat pentru a retine suma preturilor rezervarilor
        -- pentru fiecare client
        TYPE preturi_clienti IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
        preturi_clienti_rez preturi_clienti;

        v_cod_hotel hotel.cod_hotel%TYPE;
        v_nr_max_rezervari NUMBER;
        v_exista_hotel NUMBER;

        exc_nr_rezervari EXCEPTION;
        exc_nume_hotel EXCEPTION;
        PRAGMA EXCEPTION_INIT(exc_nr_rezervari, -20001);
        PRAGMA EXCEPTION_INIT(exc_nume_hotel, -20002);

    BEGIN
        -- testez daca exista hotelul cu numele nume_hotel
        SELECT COUNT(*) INTO v_exista_hotel
        FROM hotel
        WHERE UPPER(nume) = UPPER(nume_hotel);

        IF v_exista_hotel = 0 THEN
            RAISE exc_nume_hotel;
        END IF;

        SELECT cod_hotel INTO v_cod_hotel
        FROM hotel
        WHERE UPPER(nume) = UPPER(nume_hotel);

```

```

IF v_cod_hotel IS NULL THEN
    RAISE exc_nume_hotel;
END IF;

-- testez daca nr_rezervari este mai mare decat 0 si mai mic sau
egal decat numarul maxim de rezervari din acel hotel
SELECT COUNT(*) INTO v_nr_max_rezervari
FROM rezervare r JOIN camera c ON r.cod_camera = c.cod_camera
WHERE c.cod_hotel = v_cod_hotel;

IF nr_rezervari <= 0 OR nr_rezervari > v_nr_max_rezervari THEN
    RAISE exc_nr_rezervari;
END IF;

-- punem datele in lista_info_clienti
FOR client_info IN (
    SELECT c.cod_client as cod_client, c.nume || ' ' || c.prenume
    as nume_complet
        FROM rezervare r JOIN client c ON r.cod_client = c.cod_client
                    JOIN camera cam ON r.cod_camera = cam.cod_camera
                    JOIN hotel h ON cam.cod_hotel = h.cod_hotel
    WHERE h.cod_hotel = v_cod_hotel
    GROUP BY c.cod_client, c.nume, c.prenume
    HAVING COUNT(*) = nr_rezervari
) LOOP
    lista_info_clienti.EXTEND;
    lista_info_clienti(lista_info_clienti.COUNT).cod_client := client_info.cod_client;
    lista_info_clienti(lista_info_clienti.COUNT).nume_client := client_info.nume_complet;
    lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire := lista_date();
    preturi_clienti_rez(client_info.cod_client) := 0;
    FOR rezervare_info IN (SELECT data_sosire, pret
                           FROM rezervare r JOIN camera c ON
r.cod_camera = c.cod_camera
                           WHERE c.cod_hotel = v_cod_hotel AND
r.cod_client = client_info.cod_client) LOOP
        lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire.EXTEND;
        lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire(lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire.COUNT) :=

```

```

rezervare_info.data_sosire;
    preturi_clienti_rez(client_info.cod_client) :=
preturi_clienti_rez(client_info.cod_client) + rezervare_info.pret;
    END LOOP;
END LOOP;

-- daca nu exista clienti care sa aiba nr_rezervari rezervari la
hotelul cu numele nume_hotel
-- atunci dam raise la no_data_found
IF lista_info_clienti.COUNT = 0 THEN
    RAISE NO_DATA_FOUND;
ELSE
    -- daca exista astfel de clienti, atunci afisam informatiile
despre acestia
        FOR i IN 1..lista_info_clienti.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE('Nume client: ' ||
lista_info_clienti(i).nume_client);
            DBMS_OUTPUT.PUT_LINE('Lista date sosire: ');
            FOR j IN 1..lista_info_clienti(i).lista_date_sosire.COUNT
LOOP

DBMS_OUTPUT.PUT_LINE(lista_info_clienti(i).lista_date_sosire(j));
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('Suma preturi rezervari: ' ||
preturi_clienti_rez(lista_info_clienti(i).cod_client));
        DBMS_OUTPUT.PUT_LINE('-----');
        END LOOP;
    END IF;
EXCEPTION
    WHEN exc_nr_rezervari THEN
        DBMS_OUTPUT.PUT_LINE('Numarul de rezervari trebuie sa fie mai
mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel
hotel! ');
    WHEN exc_nume_hotel THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista hotelul cu numele ' ||
nume_hotel || '!');
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clienti care sa aiba ' ||
nr_rezervari || ' rezervari la hotelul cu numele ' || nume_hotel || '!');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Avem alta eroare! ');

END afisare_rezervari_hotel_ex6;

```

```

PROCEDURE rezervari_categorie_pret(categ_rezervare VARCHAR2) IS
    TYPE refcursor_rezervari IS REF CURSOR;
    rezervari_categ refcursor_rezervari;
    CURSOR c_client_hotel(cod_client_param client.cod_client%TYPE,
cod_hotel_param hotel.cod_hotel%TYPE) IS
        SELECT c.nume || ' ' || c.prenume AS nume_client, COUNT(*) AS
nr_rezervari, SUM(r.pret) AS pret_total
        FROM rezervare r JOIN client c ON r.cod_client = c.cod_client
                JOIN camera cam ON r.cod_camera =
cam.cod_camera
                JOIN hotel h ON cam.cod_hotel = h.cod_hotel
        WHERE h.cod_hotel = cod_hotel_param AND c.cod_client =
cod_client_param
        GROUP BY c.nume, c.prenume;

    v_cod_client client.cod_client%TYPE;
    v_cod_hotel hotel.cod_hotel%TYPE;
    v_pret rezervare.pret%TYPE;
    v_nume_client client.nume%TYPE;
    v_nr_rezervari NUMBER;
    v_nr_rezervari_nou NUMBER;
    v_pret_total NUMBER;

    -- tablou indexat in care sa retin pentru fiecare client
    -- care este hotelul cu rezervare din categoria data la care are
    cele mai multe rezervari
    TYPE tablou_client_hotel_max IS TABLE OF hotel.cod_hotel%TYPE INDEX
BY PLS_INTEGER;
    client_hotel_max tablou_client_hotel_max;
BEGIN
    IF LOWER(categ_rezervare) = 'ieftina' THEN
        OPEN rezervari_categ FOR
            SELECT r.cod_client, h.cod_hotel, r.pret
            FROM rezervare r JOIN camera cam ON r.cod_camera =
cam.cod_camera
                JOIN hotel h ON cam.cod_hotel =
h.cod_hotel
            WHERE r.pret < 2500;
    ELSIF LOWER(categ_rezervare) = 'medie' THEN
        OPEN rezervari_categ FOR
            SELECT r.cod_client, h.cod_hotel, r.pret

```

```

        FROM rezervare r JOIN camera cam ON r.cod_camera =
cam.cod_camera
                                JOIN hotel h ON cam.cod_hotel =
h.cod_hotel
                                WHERE r.pret >= 2500 AND r.pret <= 5000;
ELSIF LOWER(categ_rezervare) = 'scumpa' THEN
    OPEN rezervari_categ FOR
        SELECT r.cod_client, h.cod_hotel, r.pret
        FROM rezervare r JOIN camera cam ON r.cod_camera =
cam.cod_camera
                                JOIN hotel h ON cam.cod_hotel =
h.cod_hotel
                                WHERE r.pret > 5000;
ELSE
    RAISE_APPLICATION_ERROR(-20000, 'Categoria de rezervari nu este
valida!');
END IF;

LOOP
    FETCH rezervari_categ INTO v_cod_client, v_cod_hotel, v_pret;
    EXIT WHEN rezervari_categ%NOTFOUND;

    IF client_hotel_max.EXISTS(v_cod_client) THEN
        SELECT COUNT(*) INTO v_nr_rezervari
        FROM rezervare r JOIN camera cam ON (r.cod_camera =
cam.cod_camera)
                                JOIN hotel h ON (cam.cod_hotel =
h.cod_hotel)
                                WHERE r.cod_client = v_cod_client AND h.cod_hotel =
client_hotel_max(v_cod_client);

        SELECT COUNT(*) INTO v_nr_rezervari_nou
        FROM rezervare r JOIN camera cam ON (r.cod_camera =
cam.cod_camera)
                                JOIN hotel h ON (cam.cod_hotel =
h.cod_hotel)
                                WHERE r.cod_client = v_cod_client AND h.cod_hotel =
v_cod_hotel;

        IF v_nr_rezervari < v_nr_rezervari_nou THEN
            client_hotel_max(v_cod_client) := v_cod_hotel;
        END IF;
    ELSE

```

```

        client_hotel_max(v_cod_client) := v_cod_hotel;
    END IF;
END LOOP;

CLOSE rezervari_categ;

DBMS_OUTPUT.PUT_LINE('Clientii care au facut rezervari din
categoria ' || categ_rezervare || ' au istoricul:');
FOR client IN (SELECT cod_client FROM client) LOOP
    IF client_hotel_max.EXISTS(client.cod_client) THEN
        OPEN c_client_hotel(client.cod_client,
client_hotel_max(client.cod_client));
        FETCH c_client_hotel INTO v_nume_client, v_nr_rezervari,
v_pret_total;
        CLOSE c_client_hotel;

        DBMS_OUTPUT.PUT_LINE(v_nume_client || ' a facut ' ||
v_nr_rezervari || ' rezervari la hotelul ' ||
client_hotel_max(client.cod_client) || ' cu pretul total de ' ||
v_pret_total);
    END IF;
END LOOP;

END rezervari_categorie_pret;

FUNCTION lista_ang_ex8(data_param hotel.data_infintare%TYPE,
partener_param partener.nume%TYPE)
RETURN lista_nume_ex8 IS

CURSOR hoteluri_ex8 IS
    SELECT h.cod_hotel AS cod_hotel_cursor,
    CURSOR(SELECT a.nume || ' ' || a.prenume as nume_angajat
    FROM angajat a
    WHERE a.cod_hotel = h.cod_hotel
    AND a.salariu > (SELECT AVG(ang.salariu)
    FROM angajat ang
    WHERE ang.job = a.job
    GROUP BY ang.job)) AS
lista_angajati
    FROM hotel h;
sol_lista_angajati lista_nume_ex8 := lista_nume_ex8();
v_cod_hotel hotel.cod_hotel%TYPE;

```

```

v_lista_angajati SYS_REFCURSOR;
TYPE tabel_lista_angajati IS TABLE OF VARCHAR2(30) INDEX BY
BINARY_INTEGER;
tab_lista_angajati tabel_lista_angajati;

TYPE lista_coduri_hoteluri IS VARRAY(30) OF hotel.cod_hotel%TYPE;
lista_hoteluri_aprovizionate lista_coduri_hoteluri;

exc_data EXCEPTION;
PRAGMA EXCEPTION_INIT(exc_data, -20001);
exc_partener EXCEPTION;
PRAGMA EXCEPTION_INIT(exc_partener, -20002);
exc_no_data_found EXCEPTION;
PRAGMA EXCEPTION_INIT(exc_no_data_found, -20003);

data_max hotel.data_infiintare%TYPE;
v_partener_param NUMBER;

BEGIN
    SELECT MAX(data_infiintare) INTO data_max FROM hotel;
    -- verific daca exista data (sa fie mai mic sau egala cu data
    maxima la care s-a infiintat un hotel)
    IF data_param > data_max THEN
        RAISE exc_data;
    END IF;

    -- verific daca exista partenerul
    SELECT COUNT(*) INTO v_partener_param
    FROM partener
    WHERE UPPER(nume) = UPPER(partener_param);
    IF v_partener_param = 0 THEN
        RAISE exc_partener;
    END IF;

    -- retin toate hotelurile care au fost aprovizionate de partenerul
    respectiv
    SELECT h.cod_hotel BULK COLLECT INTO lista_hoteluri_aprovizionate
    FROM hotel h JOIN aprovizionare ap ON h.cod_hotel = ap.cod_hotel
                JOIN partener p ON ap.cod_partener = p.cod_partener
    WHERE UPPER(p.nume) = UPPER(partener_param) AND h.data_infiintare >
    data_param;

    -- verific daca exista vreun hotel cu conditiile date

```

```

        IF lista_hoteluri_aprovizionate.COUNT = 0 THEN
            RAISE exc_no_data_found;
        END IF;

        OPEN hoteluri_ex8;
        LOOP
            FETCH hoteluri_ex8 INTO v_cod_hotel, v_lista_angajati;
            EXIT WHEN hoteluri_ex8%NOTFOUND;
            FETCH v_lista_angajati BULK COLLECT INTO tab_lista_angajati;
            -- verifica daca v_cod_hotel se afla in lista hotelurilor
            -- aflata mai sus
            FOR i IN 1 .. lista_hoteluri_aprovizionate.COUNT LOOP
                IF v_cod_hotel = lista_hoteluri_aprovizionate(i) THEN
                    FOR j IN 1 .. tab_lista_angajati.COUNT LOOP
                        sol_lista_angajati.EXTEND;
                        sol_lista_angajati(sol_lista_angajati.COUNT) := tab_lista_angajati(j);
                    END LOOP;
                END IF;
            END LOOP;
        END LOOP;
        CLOSE hoteluri_ex8;

        RETURN sol_lista_angajati;
    EXCEPTION
        WHEN exc_data THEN
            DBMS_OUTPUT.PUT_LINE('Nu s-a infiintat niciun hotel dupa acea
data! ');
            RETURN NULL;
        WHEN exc_partener THEN
            DBMS_OUTPUT.PUT_LINE('Partenerul introdus nu exista! ');
            RETURN NULL;
        WHEN exc_no_data_found THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista hoteluri care sa satisfaca
conditiile! ');
            RETURN NULL;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Alta eroare: ' || SQLERRM);
            RETURN NULL;
    END lista_ang_ex8;

```

PROCEDURE afisare_clienti_hoteluri_piscina(cost_total_param IN NUMBER,

```

nume_client_param OUT VARCHAR2) IS

    TYPE tablou_clienti_piscina IS TABLE OF NUMBER INDEX BY
PLS_INTEGER;
        clienti_piscina tablou_clienti_piscina := tablou_clienti_piscina();

    TYPE tablou_clienti_ceruti IS TABLE OF client.cod_client%TYPE;
        clienti_ceruti tablou_clienti_ceruti := tablou_clienti_ceruti();

    TYPE tablou_clienti_rez_min IS TABLE OF client.cod_client%TYPE;
        clienti_rez_min tablou_clienti_rez_min := tablou_clienti_rez_min();

    v_client_max_p client.cod_client%TYPE;
    v_val_client_max_p NUMBER;
    v_nr_rez_min NUMBER := 3000;
    v_nume_client_aux VARCHAR2(30);
    v_nr_hoteluri_piscina NUMBER;

    exc_no_data_found EXCEPTION;
    PRAGMA EXCEPTION_INIT(exc_no_data_found, -20001);
    exc_too_many_rows EXCEPTION;
    PRAGMA EXCEPTION_INIT(exc_too_many_rows, -20002);

BEGIN

    SELECT rez_cod_client, nr_rezervari INTO v_client_max_p,
v_val_client_max_p
        FROM (
            SELECT DISTINCT r.cod_client as rez_cod_client, c.nume,
COUNT(*) AS nr_rezervari
                FROM rezervare r JOIN client c ON r.cod_client = c.cod_client
                                JOIN camera cam ON r.cod_camera =
cam.cod_camera
                                JOIN hotel h ON cam.cod_hotel = h.cod_hotel
            GROUP BY r.cod_client, c.nume
            ORDER BY nr_rezervari DESC, c.nume)
        FETCH FIRST 1 ROWS ONLY;

    -- retinem pentru fiecare client cate rezervari are la hotelurile
cu piscina
        FOR rezervare IN (   SELECT DISTINCT r.cod_client as rez_cod_client,
COUNT(*) AS nr_rezervari
                                FROM rezervare r JOIN client c ON r.cod_client

```

```

= c.cod_client
JOIN camera cam ON
r.cod_camera = cam.cod_camera
JOIN hotel h ON cam.cod_hotel
= h.cod_hotel
JOIN anexa a ON h.cod_hotel =
a.cod_hotel
WHERE UPPER(a.nume) = 'PISCINA'
GROUP BY r.cod_client
ORDER BY nr_rezervari DESC) LOOP
clienti_piscina(rezervare.rez_cod_client) :=
rezervare.nr_rezervari;
END LOOP;

-- acum selectam clientii ceruti
SELECT COUNT(DISTINCT h.cod_hotel) INTO v_nr_hoteluri_piscina
FROM rezervare r JOIN client c on (r.cod_client = c.cod_client)
JOIN camera cam on (r.cod_camera = cam.cod_camera)
JOIN hotel h on (cam.cod_hotel = h.cod_hotel)
JOIN anexa a on (h.cod_hotel = a.cod_hotel)
WHERE UPPER(a.nume) = 'PISCINA' AND r.cod_client = v_client_max_p;

SELECT c.cod_client
BULK COLLECT INTO clienti_ceruti
FROM rezervare r JOIN client c on (r.cod_client = c.cod_client)
JOIN camera cam on (r.cod_camera = cam.cod_camera)
JOIN hotel h on (cam.cod_hotel = h.cod_hotel)
JOIN anexa a on (h.cod_hotel = a.cod_hotel)
WHERE UPPER(a.nume) = 'PISCINA' AND h.cod_hotel IN ( SELECT
h1.cod_hotel
FROM rezervare
r1 JOIN client c1 on (r1.cod_client = c1.cod_client)

JOIN camera cam1 on (r1.cod_camera = cam1.cod_camera)

JOIN hotel h1 on (cam1.cod_hotel = h1.cod_hotel)

JOIN anexa a1 on (h1.cod_hotel = a1.cod_hotel)
WHERE
UPPER(a1.nume) = 'PISCINA' AND r1.cod_client = v_client_max_p)
GROUP BY c.cod_client
HAVING COUNT(DISTINCT h.cod_hotel) = v_nr_hoteluri_piscina
AND SUM(r.pret) > cost_total_param;

```

```

-- verificam daca nu exista astfel de clienti
IF SQL%ROWCOUNT = 0 THEN
    RAISE exc_no_data_found;
    --RAISE NO_DATA_FOUND;
END IF;

-- afisam clientii ceruti
FOR i IN clienti_ceruti.FIRST .. clienti_ceruti.LAST LOOP
    SELECT c.nume || ' ' || c.prenume INTO v_nume_client_aux
    FROM client c
    WHERE c.cod_client = clienti_ceruti(i);
    DBMS_OUTPUT.PUT_LINE(v_nume_client_aux);
END LOOP;

-- acum ne intereseaza clientul / clientii cu cele mai putine rezervari
FOR i IN clienti_ceruti.FIRST .. clienti_ceruti.LAST LOOP
    IF clienti_piscina(clienti_ceruti(i)) < v_nr_rez_min THEN
        v_nr_rez_min := clienti_piscina(clienti_ceruti(i));
    END IF;
END LOOP;

FOR i IN clienti_ceruti.FIRST .. clienti_ceruti.LAST LOOP
    IF clienti_piscina(clienti_ceruti(i)) = v_nr_rez_min THEN
        clienti_rez_min.EXTEND;
        clienti_rez_min(clienti_rez_min.LAST) := clienti_ceruti(i);
    END IF;
END LOOP;

-- acum selectam clientii cu cele mai putine rezervari
IF clienti_rez_min.COUNT = 1 THEN
    SELECT c.nume || ' ' || c.prenume INTO nume_client_param
    FROM client c
    WHERE c.cod_client = clienti_rez_min(1);
ELSE
    RAISE exc_too_many_rows;
    --RAISE TOO_MANY_ROWS;
END IF;

EXCEPTION
    WHEN exc_no_data_found THEN

```

```

        nume_client_param := NULL;
        DBMS_OUTPUT.PUT_LINE('Nu exista clienti care sa satisfaca
cerintele!');

        WHEN exc_too_many_rows THEN
            nume_client_param := NULL;
            DBMS_OUTPUT.PUT_LINE('Prea multi clienti cu nr minim!');

        WHEN OTHERS THEN
            nume_client_param := NULL;
            DBMS_OUTPUT.PUT_LINE('Eroare: ' || SQLERRM);

    END afisare_clienti_hoteluri_piscina;
END hoteluri_pachet_ex13;
/

BEGIN
    BEGIN
        afisare_rezervari_hotel_ex6(5, 'Vega');
        afisare_rezervari_hotel_ex6(4, 'Vega');
        afisare_rezervari_hotel_ex6(2, 'Novotel');
        afisare_rezervari_hotel_ex6(1, 'Novotel');
        afisare_rezervari_hotel_ex6(-1, 'Novotel');
        afisare_rezervari_hotel_ex6(20, 'Novotel');
        afisare_rezervari_hotel_ex6(2, 'Hotel1');
    END;

    DBMS_OUTPUT.PUT_LINE('*****');

    DECLARE
        v_categ VARCHAR2(10) := &p_categ;
    BEGIN
        rezervari_categorie_pret(v_categ);
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Exceptie: ' || SQLERRM);
    END;

    DBMS_OUTPUT.PUT_LINE('*****');

    DECLARE
        TYPE lista_angajati IS VARRAY(30) OF VARCHAR2(30);
        solutie lista_numa_ex8;
    BEGIN
        solutie := lista_ang_ex8(TO_DATE('01-01-2010', 'DD-MM-YYYY'),

```

```

'Lidl');
    IF solutie IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Lista angajatilor: ');
        FOR i IN 1 .. solutie.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE(solutie(i));
        END LOOP;
    END IF;

    DBMS_OUTPUT.PUT_LINE('-----');

    solutie := lista_ang_ex8(TO_DATE('01-01-2025', 'DD-MM-YYYY'),
'Lidl');
    IF solutie IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Lista angajatilor: ');
        FOR i IN 1 .. solutie.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE(solutie(i));
        END LOOP;
    END IF;

    DBMS_OUTPUT.PUT_LINE('-----');

    solutie := lista_ang_ex8(TO_DATE('01-01-2010', 'DD-MM-YYYY'),
'IKEA');
    IF solutie IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Lista angajatilor: ');
        FOR i IN 1 .. solutie.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE(solutie(i));
        END LOOP;
    END IF;

    DBMS_OUTPUT.PUT_LINE('-----');

    INSERT INTO hotel VALUES (99, 1, 'Hilton', TO_DATE('01-01-2010',
'DD-MM-YYYY'), 65, 5);
    INSERT INTO aprovizionare VALUES (12, 99, TO_DATE('01-01-2015',
'DD-MM-YYYY'), 200);
    solutie := lista_ang_ex8(TO_DATE('31-12-2024', 'DD-MM-YYYY'),
'Cif');
    IF solutie IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Lista angajatilor: ');
        FOR i IN 1 .. solutie.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE(solutie(i));
        END LOOP;
    END IF;

```

```

        END IF;

        ROLLBACK;
    END;

    DBMS_OUTPUT.PUT_LINE('*****');

    DECLARE
        v_nume_client VARCHAR2(30);
    BEGIN
        afisare_clienti_hoteluri_piscina(1000, v_nume_client);
        IF v_nume_client IS NOT NULL THEN
            DBMS_OUTPUT.PUT_LINE('Clientul cu cele mai putine rezervari: ' ||
|| v_nume_client);
            END IF;

            DBMS_OUTPUT.PUT_LINE('-----');

            afisare_clienti_hoteluri_piscina(10000, v_nume_client);
            IF v_nume_client IS NOT NULL THEN
                DBMS_OUTPUT.PUT_LINE('Clientul cu cele mai putine rezervari: ' ||
|| v_nume_client);
                END IF;

                DBMS_OUTPUT.PUT_LINE('-----');

                INSERT INTO client VALUES (1000, 'Popescu', 'Ion', '0722222222');
                INSERT INTO rezervare VALUES (1000, 103, 149, TO_DATE('24-08-2020',
'DD-MM-YYYY'), 2000);
                afisare_clienti_hoteluri_piscina(1000, v_nume_client);
                IF v_nume_client IS NOT NULL THEN
                    DBMS_OUTPUT.PUT_LINE('Clientul cu cele mai putine rezervari: ' ||
|| v_nume_client);
                    END IF;
                    ROLLBACK;

    END;

END;
/

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Project:** sqlproject
- Console:** console_7 [project_sgbdb]
- Code Editor:** The code is a PL/SQL procedure named `afisare_clienti_hoteluri_piscina`. It includes logic to handle categories and output results.
- Parameter Substitution Dialog:** A modal dialog is displayed, asking for a value for the parameter `v_categ`. The placeholder value shown is `MEDIE`.
- Buttons:** The dialog contains "Execute" and "Close" buttons.
- Code Preview:** Below the editor, a preview pane shows the resulting SQL or PL/SQL code after the parameter is substituted.

```
413      END afisare_clienti_hoteluri_piscina;
414
415  END hoteluri_pachet_ex13;
416
417
418  BEGIN
419    BEGIN
420      afisare_rezervari_hotel_ex
421      afisare_rezervari_hotel_e
422      afisare_rezervari_hotel_e
423      afisare_rezervari_hotel_e
424      afisare_rezervari_hotel_e
425      afisare_rezervari_hotel_e
426      afisare_rezervari_hotel_e
427      afisare_rezervari_hotel_e
428    END;
429
430    DBMS_OUTPUT.PUT_LINE( A: '*****');
431
432    DECLARE
433      v_categ VARCHAR2(10) := &v_categ;
434
435    BEGIN
436      rezervari_categorie_pret( CATEG_REZERVARE v_categ);
437      EXCEPTION
438        WHEN OTHERS THEN
439          DBMS_OUTPUT.PUT_LINE( A: 'Excepție: ' || SQLERRM);
440    END;
441
442    DBMS_OUTPUT.PUT_LINE( A: '*****');
443
444    DECLARE
445      TYPE lista_angajati IS VARRAY(30) OF VARCHAR2(30);
446      solutie lista_nume_ex8;
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- File Explorer:** Shows the project structure under "Scratches and Consoles". The "project_sgbd" scratch contains a script named "console_7.sql".
- Editor:** The "console_7.sql" script is open, containing a PL/SQL block that inserts 10 rows into the "afisare_rezervari_hotel_ex6" table and then prints a separator line and declares a variable.
- Log:** The log pane shows the execution results:
 - [2024-01-08 22:41:10] completed in 15 ms
 - Nu există clienti care să aibă 5 rezervări la hotelul cu numele Vega!
 - Nume client: Vasilescu Andrei
 - Lista date sosire:
 - 23-NOV-19
 - 19-MAR-21
 - 05-JUN-20
 - 27-OCT-21
 - Sumă prețuri rezervări: 16600
- Database Consoles:** Shows the execution history for "project_sgbd":
 - console_9 (99 ms)
 - console_8 (99 ms)
 - console_7 (99 ms)
 - console_6 (126 ms)
 - console_4 (126 ms)
 - console_7 (58 ms)Each execution shows the same result as the log.
- Bottom Bar:** Includes icons for file operations, search, and various developer tools like Git, Taskbar, and Help.

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\Scratches and Consoles\Database Consoles\project\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_9 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_8 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_6 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_4 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_3 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_2 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_1 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console [project_sgbd]
- Scratches and Consoles\vlad\console [vlad]

Services

Tx

- Database\project_sgbd\console_9
- Database\project_sgbd\console_8
- Database\project_sgbd\console
- Database\project_sgbd\console_4
- Database\project_sgbd\console_6
- Database\project_sgbd\console_7
- Database\project_sgbd\console_4
- Database\project_sgbd\console_7

```

418 BEGIN
419   afisare_rezervari_hotel_ex6( NR_REZERVARI: 5, NUME_HOTEL: 'Vega');
420   afisare_rezervari_hotel_ex6( NR_REZERVARI: 4, NUME_HOTEL: 'Vega');
421   afisare_rezervari_hotel_ex6( NR_REZERVARI: 2, NUME_HOTEL: 'Novotel');
422   afisare_rezervari_hotel_ex6( NR_REZERVARI: 1, NUME_HOTEL: 'Novotel');
423   afisare_rezervari_hotel_ex6( NR_REZERVARI: 1, NUME_HOTEL: 'Novotel');
424   afisare_rezervari_hotel_ex6( NR_REZERVARI: 20, NUME_HOTEL: 'Novotel');
425   afisare_rezervari_hotel_ex6( NR_REZERVARI: 2, NUME_HOTEL: 'Hotell');
426
427   END;
428
429   DBMS_OUTPUT.PUT_LINE( A: '*****');
430
431   DECLARE
432     v_categ VARCHAR2(10) := &p_categ;
433   BEGIN
434   END;
435
436   RERTRN

```

Suma preturi rezervari: 1800

Numarul de rezervari trebuie sa fie mai mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel hotel!

Numarul de rezervari trebuie sa fie mai mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel hotel!

Nu exista hotelul cu numele Hotell!

Clienții care au făcut rezervari din categoria MEDIE au istoricul:

Stoica Roxana a făcut 2 rezervari la hotelul 8 cu prețul total de 6300

Vasilescu Andrei a făcut 4 rezervari la hotelul 10 cu prețul total de 16600

418:6 CRLF UTF-8 4 spaces 10:42 PM 1/8/2024

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\Scratches and Consoles\Database Consoles\project\project_sgbd\console_7 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_9 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_8 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_6 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_4 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_3 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_2 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console_1 [project_sgbd]
- Scratches and Consoles\Database Consoles\project\project_sgbd\console [project_sgbd]
- Scratches and Consoles\vlad\console [vlad]

Services

Tx

- Database\project_sgbd\console_9
- Database\project_sgbd\console_8
- Database\project_sgbd\console
- Database\project_sgbd\console_4
- Database\project_sgbd\console_6
- Database\project_sgbd\console_7
- Database\project_sgbd\console_4
- Database\project_sgbd\console_7

```

418 BEGIN
419   afisare_rezervari_hotel_ex6( NR_REZERVARI: 5, NUME_HOTEL: 'Vega');
420   afisare_rezervari_hotel_ex6( NR_REZERVARI: 4, NUME_HOTEL: 'Vega');
421   afisare_rezervari_hotel_ex6( NR_REZERVARI: 2, NUME_HOTEL: 'Novotel');
422   afisare_rezervari_hotel_ex6( NR_REZERVARI: 1, NUME_HOTEL: 'Novotel');
423   afisare_rezervari_hotel_ex6( NR_REZERVARI: 20, NUME_HOTEL: 'Novotel');
424   afisare_rezervari_hotel_ex6( NR_REZERVARI: 2, NUME_HOTEL: 'Hotell');
425
426   END;
427
428   DBMS_OUTPUT.PUT_LINE( A: '*****');
429
430   DECLARE
431     v_categ VARCHAR2(10) := &p_categ;
432   BEGIN
433   END;
434
435   RERTRN

```

Stoica Roxana a făcut 2 rezervari la hotelul 8 cu prețul total de 6300

Vasilescu Andrei a făcut 4 rezervari la hotelul 10 cu prețul total de 16600

Lista angajaților:

Popescu Alexandru

Rădulescu Luca

Avrăm Cristian

Abu Victor

Stanescu Ana

Enache Corina

Constantinescu Diana

Pop Adrian

Ionescu Radu

Winter David

418:6 CRLF UTF-8 4 spaces 10:42 PM 1/8/2024

The screenshot shows a SQL development environment with two panes. The left pane displays a file tree under 'sqlproject' containing several database consoles, including 'console_7 [project_sgbd]'. The right pane shows the code for a stored procedure named 'afisare_rezervari_hotel_ex6'. The code includes a BEGIN block with multiple calls to 'afisare_rezervari_hotel_ex6' with specific parameters (NR_REZERVARE values 5, 4, 2, 1, 20, 2) and a corresponding END block. Below the code, there is a DBMS_OUTPUT.PUT_LINE statement that prints a separator line. A 'DECLARE' section follows, defining a variable 'v_categ' of type VARCHAR2(10) assigned to '&p_categ'. The final line is 'RETURNS'. The bottom pane, titled 'Services', shows a transaction list with various users and their connection times. One entry for 'console_7' is highlighted with a red border and shows a duration of 58 ms. The status bar at the bottom indicates the date and time as 1/8/2024, 10:42 PM.

```
418 ✓ BEGIN
419     BEGIN
420         afisare_rezervari_hotel_ex6( NR_REZERVARE 5, NUME_HOTEL: 'Vega');
421         afisare_rezervari_hotel_ex6( NR_REZERVARE 4, NUME_HOTEL: 'Vega');
422         afisare_rezervari_hotel_ex6( NR_REZERVARE 2, NUME_HOTEL: 'Novotel');
423         afisare_rezervari_hotel_ex6( NR_REZERVARE 1, NUME_HOTEL: 'Novotel');
424         afisare_rezervari_hotel_ex6( NR_REZERVARE 1, NUME_HOTEL: 'Novotel');
425         afisare_rezervari_hotel_ex6( NR_REZERVARE 20, NUME_HOTEL: 'Novotel');
426         afisare_rezervari_hotel_ex6( NR_REZERVARE 2, NUME_HOTEL: 'Hotell');
427
428     END;
429
430     DBMS_OUTPUT.PUT_LINE( A: '*****');
431
432     DECLARE
433         v_categ VARCHAR2(10) := &p_categ;
434
435     RETURNS
```

This screenshot is nearly identical to the one above, showing the same SQL environment and code for the 'afisare_rezervari_hotel_ex6' stored procedure. The main difference is in the 'Services' pane, where the transaction list now includes additional names such as 'Tomas Radu', 'Calinescu Cosmin', 'Zaharia Marius', 'Toma Andrei', 'Stanciu Stefan', 'Munteanu Mihai', 'Neagu Dana', 'Cojocaru Cosmina', 'Ionita Irina', and 'Dinu Ana'. The transaction for 'console_7' is still highlighted with a red border and shows a duration of 58 ms. The status bar at the bottom indicates the date and time as 1/8/2024, 10:42 PM.

```
418 ✓ BEGIN
419     BEGIN
420         afisare_rezervari_hotel_ex6( NR_REZERVARE 5, NUME_HOTEL: 'Vega');
421         afisare_rezervari_hotel_ex6( NR_REZERVARE 4, NUME_HOTEL: 'Vega');
422         afisare_rezervari_hotel_ex6( NR_REZERVARE 2, NUME_HOTEL: 'Novotel');
423         afisare_rezervari_hotel_ex6( NR_REZERVARE 1, NUME_HOTEL: 'Novotel');
424         afisare_rezervari_hotel_ex6( NR_REZERVARE 20, NUME_HOTEL: 'Novotel');
425         afisare_rezervari_hotel_ex6( NR_REZERVARE 2, NUME_HOTEL: 'Hotell');
426
427     END;
428
429     DBMS_OUTPUT.PUT_LINE( A: '*****');
430
431     DECLARE
432         v_categ VARCHAR2(10) := &p_categ;
433
434     RETURNS
```

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\sqlproject
- Scratches and Consoles
 - Database Consoles
 - project
 - project_sgbd
 - console [project_sgbd]
 - console_1 [project_sgbd]
 - console_2 [project_sgbd]
 - console_3 [project_sgbd]
 - console_4 [project_sgbd]
 - console_5 [project_sgbd]
 - console_6 [project_sgbd]
 - console_7 [project_sgbd]
 - console_8 [project_sgbd]
 - console_9 [project_sgbd]
 - vlad
 - console [vlad]

Services

Tx

- Database
 - project_sgbd
 - console_9
 - console_8
 - console 99 ms
 - console 99 ms
 - console 99 ms
 - console_6
 - console_4 126 ms
 - console_4 126 ms
 - console_4 126 ms
 - console_7 58 ms

Database Consoles > project_sgbd > console_7 [project_sgbd]

```

418 BEGIN
419     afisare_rezervari_hotel_ex6( NR_REZERVARE_5, NUME_HOTEL: 'Vega');
420     afisare_rezervari_hotel_ex6( NR_REZERVARE_4, NUME_HOTEL: 'Vega');
421     afisare_rezervari_hotel_ex6( NR_REZERVARE_2, NUME_HOTEL: 'Novotel');
422     afisare_rezervari_hotel_ex6( NR_REZERVARE_1, NUME_HOTEL: 'Novotel');
423     afisare_rezervari_hotel_ex6( NR_REZERVARE_1, NUME_HOTEL: 'Novotel');
424     afisare_rezervari_hotel_ex6( NR_REZERVARE_29, NUME_HOTEL: 'Novotel');
425     afisare_rezervari_hotel_ex6( NR_REZERVARE_2, NUME_HOTEL: 'Hotell');
426
427     DBMS_OUTPUT.PUT_LINE( A || '*****');
428
429     DECLARE
430         v_categ VARCHAR2(10) := &p_categ;
431     BEGIN
432     END;
433 
```

418:6 CRLF UTF-8 4 spaces 10:42 PM 1/8/2024

sqlproject Version control

Files

- sqlproject C:\Users\vcupu\Documents\sqlproject
- Scratches and Consoles
 - Database Consoles
 - project
 - project_sgbd
 - console [project_sgbd]
 - console_1 [project_sgbd]
 - console_2 [project_sgbd]
 - console_3 [project_sgbd]
 - console_4 [project_sgbd]
 - console_5 [project_sgbd]
 - console_6 [project_sgbd]
 - console_7 [project_sgbd]
 - console_8 [project_sgbd]
 - console_9 [project_sgbd]
 - vlad
 - console [vlad]

Services

Tx

- Database
 - project_sgbd
 - console_9
 - console_8
 - console 99 ms
 - console 99 ms
 - console 99 ms
 - console_6
 - console_4 126 ms
 - console_4 126 ms
 - console_4 126 ms
 - console_7 89 ms

Database Consoles > project_sgbd > console_7 [project_sgbd]

```

428 DBMS_OUTPUT.PUT_LINE( A || '*****');
429
430     DECLARE
431         v_categ VARCHAR2(10) := &p_categ;
432     BEGIN
433         rezervari_categorie_pret( CATEG.REZERVARE: v_categ);
434     EXCEPTION
435         WHEN OTHERS THEN
436             DBMS_OUTPUT.PUT_LINE( A || 'Excepție: ' || SQLERRM);
437     END;
438
439     DBMS_OUTPUT.PUT_LINE( A || '*****');
440
441     DECLARE
442         TYPE lista_angajatilor IS VARRAY(30) OF VARCHAR2(30);
443 
```

NUMARUL DE REZERVARI TREBUIE SA TIE MAI MARE DECAT SI MAI MIC SAU EGAL DECAT NUMARUL MAXIM DE REZERVARI DIN ACEL HOTEL!
 Nu exista hotelul cu numele Hotel!

Excepție: ORA-20000: Categoria de rezervari nu este validă!

Lista angajatilor:

Popescu Alexandru
 Radulescu Luca
 Avram Cristian
 Albu Victor
 Stanescu Ana
 Enache Corina
 Constantinescu Diana

10:44 PM 1/8/2024

14. Definiți 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).

```
/* In fiecare trimestru, se stabileste performanta hotelurilor fata de anul precedent, in aceeasi perioada, prin calcularea incasarilor totale avute. Sa se verifice daca incasarile totale pe ultimul an in care s-au inregistrat rezervari sunt mai mari decat incasarile totale pe anul precedent, raportat la trimestrul in care ne aflam in prezent (fiecare trimestru fiind codificat prin 1, 2, 3, in ordinea acestora din an).
```

In caz afirmativ, in cadrul hotelurilor in care s-au facut cel putin atatea rezervari cat este media numarului de rezervari per hotel (in cazul in care media nu va fi un numar intreg, se va lua partea intreaga a acesteia), sa se mareasca cu 500 de lei salariul fiecarui angajat care era platit mai putin de 20% din salariul managerului sau. De asemenea, sa se afiseze numele angajatilor care au primit marire de salariu, precum si noul salariu al acestora.

In caz negativ, in care am avut incasari mai mici, sa se scada cu 30 de lei preturile pachetelor oferite de hotelurile care au avut cel mult atatea rezervari cat media calculata anterior. De asemenea, sa se afiseze numele hotelurilor care au scazut pretul pachetelor, precum si noul pret al acestora.

In cazul in care incasarile sunt egale (performanta este aceeasi), fiecare manager care are macar un angajat cu salariul mai mare de 6000 de lei, sa primeasca un bonus la salariu egal cu 10% din salariul angajatului pe care il au in subordine cu cel mai mic salariu. De asemenea, sa se afiseze numele managerilor care au primit bonus, precum si noul salariu al acestora.

```
*/
```

Rezolvare:

```
CREATE OR REPLACE PACKAGE hoteluri_pachet_ex14 IS  
  
    FUNCTION trimestru_curent RETURN NUMBER;  
    PROCEDURE cei_mai_recenti_ani_inregistrati(an1 OUT NUMBER, an2 OUT  
NUMBER);  
  
    TYPE t_lin IS TABLE OF NUMBER INDEX BY PLS_INTEGER;  
    TYPE t_mat IS TABLE OF t_lin INDEX BY PLS_INTEGER;
```

```

mat_incasari t_mat := t_mat();
PROCEDURE init_incasari_trimestriale(an NUMBER, trimestru NUMBER);

TYPE info_ang IS RECORD(
    cod_angajat angajat.cod_angajat%TYPE,
    nume_complet VARCHAR2(30),
    salariu angajat.salariu%TYPE
);
TYPE tab_info_ang IS TABLE OF info_ang;
FUNCTION lista_angajati_eligibili_pentru_marire(cod_hotel_dorit
hotel.cod_hotel%TYPE) RETURN tab_info_ang;

FUNCTION hoteluri_cu_modificari(tip_cautare NUMBER) RETURN
SYS_REFCURSOR;

PROCEDURE marire_salarii_angajati(cod_hotel_dorit
hotel.cod_hotel%TYPE);
PROCEDURE scadere_preturi_pachete(cod_hotel_dorit
hotel.cod_hotel%TYPE);
PROCEDURE bonus_manageri;

PROCEDURE verificare_trimestriala;
END hoteluri_pachet_ex14;

```

```

CREATE OR REPLACE PACKAGE BODY hoteluri_pachet_ex14 IS

FUNCTION trimestru_curent RETURN NUMBER IS
    luna_curenta NUMBER;
BEGIN
    SELECT EXTRACT(MONTH FROM SYSDATE) INTO luna_curenta
    FROM DUAL;

CASE
    WHEN luna_curenta BETWEEN 1 AND 4 THEN
        RETURN 1;
    WHEN luna_curenta BETWEEN 5 AND 8 THEN
        RETURN 2;
    WHEN luna_curenta BETWEEN 9 AND 12 THEN
        RETURN 3;
END CASE;
-- calcul direct
--RETURN (luna_curenta - 1) / 4 + 1;

```

```

END trimestru_curent;

PROCEDURE cei_mai_recenti_ani_inregistrati(an1 OUT NUMBER, an2 OUT
NUMBER) IS
BEGIN
    -- stim deja ca lantul hotelier exista de mai mult de 2 ani
    SELECT MAX(EXTRACT(YEAR FROM data_sosire)) INTO an1
    FROM rezervare
    FETCH FIRST ROW ONLY;

    SELECT MAX(EXTRACT(YEAR FROM data_sosire)) INTO an2
    FROM rezervare
    WHERE EXTRACT(YEAR FROM data_sosire) <> an1
    FETCH FIRST ROW ONLY;
END cei_mai_recenti_ani_inregistrati;

PROCEDURE init_incasari_trimestriale(an NUMBER, trimestru NUMBER) IS
BEGIN
    SELECT SUM(pret) INTO mat_incasari(an)(trimestru)
    FROM rezervare
    WHERE EXTRACT(YEAR FROM data_sosire) = an
        AND (EXTRACT(MONTH FROM data_sosire) BETWEEN (trimestru - 1) *
4 + 1 AND trimestru * 4);

    IF mat_incasari(an)(trimestru) IS NULL THEN
        RAISE NO_DATA_FOUND;
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        mat_incasari(an)(trimestru) := -1;
        --DBMS_OUTPUT.PUT_LINE('Nu exista rezervari in trimestrul ' ||
trimestru || ' al anului ' || an);
    END init_incasari_trimestriale;

FUNCTION lista_angajati_eligibili_pentru_marire(cod_hotel_dorit
hotel.cod_hotel%TYPE) RETURN tab_info_ang IS
    lista_info_ang tab_info_ang;
BEGIN
    SELECT a.cod_angajat, a.nume || ' ' || a.prenume, a.salariu BULK
COLLECT INTO lista_info_ang
        FROM angajat a JOIN angajat man ON (a.cod_manager =
man.cod_angajat)

```

```

        WHERE a.cod_manager IS NOT NULL AND a.cod_hotel = cod_hotel_dorit
              AND a.salariu < man.salariu * 0.2;

        IF lista_info_ang.COUNT = 0 THEN
            RAISE NO_DATA_FOUND;
        END IF;

        RETURN lista_info_ang;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista angajati eligibili pentru
marire');
        RETURN NULL;

END lista_angajati_eligibili_pentru_marire;

FUNCTION hoteluri_cu_modificari(tip_cautare NUMBER) RETURN
SYS_REF_CURSOR IS
    c SYS_REFCURSOR;
    nr_mediul_rezervari NUMBER;
    exc_tip_cautare EXCEPTION;
    PRAGMA EXCEPTION_INIT(exc_tip_cautare, -20001);
BEGIN
    IF tip_cautare <> -1 AND tip_cautare <> 1 THEN
        RAISE exc_tip_cautare;
    END IF;

    SELECT FLOOR(AVG(COUNT(*))) INTO nr_mediul_rezervari
    FROM rezervare r JOIN camera cam ON (r.cod_camera = cam.cod_camera)
    GROUP BY cam.cod_hotel;

    -- SQL dinamic: tip_cautare = 1 => creste pretul, tip_cautare = -1
=> scade pretul
    -- sigur avem hoteluri eligibile pentru modificari (mereu exista
    macar un hotel sub medie si unul peste medie)
    OPEN c FOR
        'SELECT h.cod_hotel, h.nume ' ||
        'FROM rezervare r JOIN camera cam ON (r.cod_camera =
    cam.cod_camera) ' ||
        'JOIN hotel h ON (cam.cod_hotel = h.cod_hotel)
' ||
        'GROUP BY h.cod_hotel, h.nume ' ||

```

```

        ' HAVING (COUNT(*) - :nr_mediul_rezervari) * :tip_cautare >= 0 '
        USING nr_mediul_rezervari, tip_cautare;

    RETURN c;

EXCEPTION
    WHEN exc_tip_cautare THEN
        DBMS_OUTPUT.PUT_LINE('Tipul cautarii trebuie sa fie -1 sau 1');
        RETURN NULL;
END hoteluri_cu_modificari;

PROCEDURE marire_salarii_angajati(cod_hotel_dorit hotel.cod_hotel%TYPE)
IS
    lista_info_ang tab_info_ang;
BEGIN
    lista_info_ang := 
lista_angajati_eligibili_pentru_marire(cod_hotel_dorit);
    IF lista_info_ang IS NOT NULL THEN
        FOR i IN 1 .. lista_info_ang.COUNT LOOP
            UPDATE angajat a
            SET a.salariu = a.salariu + 500
            WHERE a.cod_angajat = lista_info_ang(i).cod_angajat;
            lista_info_ang(i).salariu := lista_info_ang(i).salariu +
500;
            DBMS_OUTPUT.PUT_LINE('Angajatul ' ||
lista_info_ang(i).nume_complet || ' a primit marire de salariu, nou
salariu fiind ' || lista_info_ang(i).salariu );
        END LOOP;
    END IF;
END marire_salarii_angajati;

PROCEDURE scadere_preturi_pachete(cod_hotel_dorit hotel.cod_hotel%TYPE)
IS
    -- fiecare hotel ofera cel putin un pachet
    CURSOR pachetele_hotelului IS
        SELECT cod_pachet, beneficiu, pret
        FROM pachet
        WHERE cod_hotel = cod_hotel_dorit;
        v_pret_actualizat pachet.pret%TYPE;
BEGIN
    FOR pachetul_curent IN pachetele_hotelului LOOP
        UPDATE pachet p
        SET p.pret = p.pret - 30
    END LOOP;
END scadere_preturi_pachete;

```

```

        WHERE p.cod_pachet = pachetul_curent.cod_pachet;
        v_pret_actualizat := pachetul_curent.pret - 30;
        DBMS_OUTPUT.PUT_LINE('Pachetul cu beneficiul ' ||
pachetul_curent.beneficiu || ' a scazut pretul la ' || v_pret_actualizat);
    END LOOP;
END scadere_preturi_pachete;

PROCEDURE bonus_manageri IS
    TYPE info_manager IS RECORD(
        cod_angajat angajat.cod_angajat%TYPE,
        nume_complet VARCHAR2(30),
        salariu_angajat.salariu%TYPE,
        salariu_minim angajat.salariu%TYPE
    );
    TYPE tab_manageri IS TABLE OF info_manager INDEX BY PLS_INTEGER;
    manageri_eligibili tab_manageri := tab_manageri();

    ok BOOLEAN;
    sal_min angajat.salariu%TYPE;

    exc_manageri EXCEPTION;
    PRAGMA EXCEPTION_INIT(exc_manageri, -20001);

BEGIN
    FOR managerul_curent IN (SELECT cod_angajat, nume || ' ' || prenume
as nume_complet, salariu
                                         FROM angajat
                                         WHERE cod_manager IS NULL) LOOP
        ok := FALSE;
        sal_min := 99999;
        FOR angajatul_curent IN (SELECT cod_angajat, salariu
                                         FROM angajat
                                         WHERE cod_manager =
managerul_curent.cod_angajat) LOOP
            IF angajatul_curent.salariu < sal_min THEN
                sal_min := angajatul_curent.salariu;
            END IF;

            IF angajatul_curent.salariu > 6000 THEN
                ok := TRUE;
            END IF;
        END LOOP;

        IF ok THEN

```

```

        manageri_eligibili(manageri_eligibili.COUNT +
1).cod_angajat := managerul_curent.cod_angajat;
        manageri_eligibili(manageri_eligibili.COUNT).nume_complet
:= managerul_curent.nume_complet;
        manageri_eligibili(manageri_eligibili.COUNT).salariu :=
managerul_curent.salariu;
        manageri_eligibili(manageri_eligibili.COUNT).salariu_minim
:= sal_min;
    END IF;
END LOOP;

IF manageri_eligibili.COUNT = 0 THEN
    RAISE exc_manageri;
END IF;

FOR i IN 1..manageri_eligibili.COUNT LOOP
    UPDATE angajat
    SET salariu = salariu + 0.1 *
manageri_eligibili(i).salariu_minim
    WHERE cod_angajat = manageri_eligibili(i).cod_angajat;
    manageri_eligibili(i).salariu := manageri_eligibili(i).salariu
+ 0.1 * manageri_eligibili(i).salariu_minim;
    DBMS_OUTPUT.PUT_LINE('Managerul ' ||
manageri_eligibili(i).nume_complet || ' a primit bonus, nou salarui fiind
' || manageri_eligibili(i).salariu);
    END LOOP;

EXCEPTION
    WHEN exc_manageri THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista manageri eligibili pentru
bonus') ;

END bonus_manageri;

PROCEDURE verificare_trimestrială IS
    an1 NUMBER; an2 NUMBER;
    cod_hotel_current hotel.cod_hotel%TYPE;
    nume_hotel_curent hotel.nume%TYPE;
    c SYS_REFCURSOR;

    exc_perioada EXCEPTION;
PRAGMA EXCEPTION_INIT(exc_perioada, -20001);

```

```

BEGIN
    cei_mai_recenti_ani_inregistrati(an1, an2);
    DBMS_OUTPUT.PUT_LINE('Ultimii ani inregistrati pe care verificam
performanta sunt: ' || an1 || ' si ' || an2);

    init_incasari_trimestriale(an1, trimestru_curent);
    init_incasari_trimestriale(an2, trimestru_curent);

    -- daca nu exista rezervari in trimestrul trecut, punand -1 in
matrice, vom avea incasarile curente mai mari
    IF mat_incasari(an1)(trimestru_curent) = -1 THEN
        RAISE exc_perioada;
    END IF;

    CASE
        WHEN mat_incasari(an1)(trimestru_curent) >
mat_incasari(an2)(trimestru_curent) THEN
            DBMS_OUTPUT.PUT_LINE('Incasarile curente sunt mai mari
decat cele din anul precedent');
            c := hoteluri_cu_modificari(1);

        LOOP
            FETCH c INTO cod_hotel_curent, nume_hotel_curent;
            EXIT WHEN c%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('In hotelul ' || ' ' ||
nume_hotel_curent || ':' );
            marire_salarii_angajati(cod_hotel_curent);
        END LOOP;

        WHEN mat_incasari(an1)(trimestru_curent) <
mat_incasari(an2)(trimestru_curent) THEN
            DBMS_OUTPUT.PUT_LINE('Incasarile curente sunt mai mici
decat cele din anul precedent');
            c := hoteluri_cu_modificari(-1);

        LOOP
            FETCH c INTO cod_hotel_curent, nume_hotel_curent;
            EXIT WHEN c%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('In hotelul ' || ' ' ||
nume_hotel_curent || ':' );
            scadere_preturi_pachete(cod_hotel_curent);
        END LOOP;

```

```

        WHEN mat_incasari(an1)(trimestru_curent) =
mat_incasari(an2)(trimestru_curent) THEN
            DBMS_OUTPUT.PUT_LINE('Incasarile curente sunt egale cu cele
din anul precedent');
            bonus_manageri;
    END CASE;

EXCEPTION
    WHEN exc_perioada THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista rezervari perioada curenta');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare: ' || SQLERRM);
END verificare_trimestriala;

END hoteluri_pachet_ex14;
/

```

```

BEGIN
    BEGIN
        DBMS_OUTPUT.PUT_LINE('Trimestrul curent este: ' ||
hoteluri_pachet_ex14.trimestru_curent);
    END;

    DBMS_OUTPUT.PUT_LINE('*****');

DECLARE
    an1 NUMBER; an2 NUMBER;
BEGIN
    hoteluri_pachet_ex14.cei_mai_recenti_ani_inregistrati(an1, an2);
    DBMS_OUTPUT.PUT_LINE('Ultimii ani inregistrati sunt: ' || an1 || ' '
si ' || an2);
    END;

    DBMS_OUTPUT.PUT_LINE('*****');

DECLARE
    an1 NUMBER := 2020;
    trimestru1 NUMBER := 1;
    an2 NUMBER := 2027;
    trimestru2 NUMBER := 3;
BEGIN
    hoteluri_pachet_ex14.init_incasari_trimestriale(an1, trimestru1);

```

```

        IF hoteluri_pachet_ex14.mat_incasari(an1)(trimestru1) <> -1 THEN
            DBMS_OUTPUT.PUT_LINE('Incasarile din trimestrul ' || trimestru1
|| ' al anului ' || an1 || ' sunt: ' ||
hoteluri_pachet_ex14.mat_incasari(an1)(trimestru1));
        ELSE
            DBMS_OUTPUT.PUT_LINE('Nu exista rezervari in trimestrul ' || trimestru1 || ' al anului ' || an1);
        END IF;

        hoteluri_pachet_ex14.init_incasari_trimestriale(an2, trimestru2);
        IF hoteluri_pachet_ex14.mat_incasari(an2)(trimestru2) <> -1 THEN
            DBMS_OUTPUT.PUT_LINE('Incasarile din trimestrul ' || trimestru2
|| ' al anului ' || an2 || ' sunt: ' ||
hoteluri_pachet_ex14.mat_incasari(an2)(trimestru2));
        ELSE
            DBMS_OUTPUT.PUT_LINE('Nu exista rezervari in trimestrul ' || trimestru2 || ' al anului ' || an2);
        END IF;
    END;

    DBMS_OUTPUT.PUT_LINE('*****');

```

BEGIN

DECLARE

lista_ang hoteluri_pachet_ex14.tab_info_ang;

BEGIN

lista_ang :=

hoteluri_pachet_ex14.lista_angajati_eligibili_pentru_marire(9);

IF lista_ang IS NOT NULL THEN

FOR i IN 1..lista_ang.COUNT LOOP

DBMS_OUTPUT.PUT_LINE(lista_ang(i).nume_complet || ' ' ||

|| lista_ang(i).salariu);

END LOOP;

END IF;

END;

DECLARE

lista_ang hoteluri_pachet_ex14.tab_info_ang;

BEGIN

UPDATE angajat

SET salariu = 99999

WHERE cod_hotel = 9 AND cod_manager IS NOT NULL;

```

        lista_ang := hoteluri_pachet_ex14.lista_angajati_eligibili_pentru_marire(9);
        IF lista_ang IS NOT NULL THEN
            FOR i IN 1..lista_ang.COUNT LOOP
                DBMS_OUTPUT.PUT_LINE(lista_ang(i).nume_complet || ' '
|| lista_ang(i).salariu);
            END LOOP;
        END IF;

        ROLLBACK;
    END;
END;

DBMS_OUTPUT.PUT_LINE('*****');

BEGIN
    DECLARE
        c SYS_REFCURSOR;
        cod_hotel_curent hotel.cod_hotel%TYPE;
        nume_hotel_curent hotel.nume%TYPE;
    BEGIN
        c := hoteluri_pachet_ex14.hoteluri_cu_modificari(1);
        IF c IS NOT NULL THEN
            LOOP
                FETCH c INTO cod_hotel_curent, nume_hotel_curent;
                EXIT WHEN c%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE(nume_hotel_curent);
            END LOOP;
        END IF;
    END;

    DECLARE
        c SYS_REFCURSOR;
        cod_hotel_curent hotel.cod_hotel%TYPE;
        nume_hotel_curent hotel.nume%TYPE;
    BEGIN
        c := hoteluri_pachet_ex14.hoteluri_cu_modificari(-1);
        IF c IS NOT NULL THEN
            LOOP
                FETCH c INTO cod_hotel_curent, nume_hotel_curent;
                EXIT WHEN c%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE(nume_hotel_curent);
            END LOOP;
        END IF;
    END;

```

```

        END IF;
    END;

    DECLARE
        c SYS_REFCURSOR;
        cod_hotel_curent hotel.cod_hotel%TYPE;
        nume_hotel_curent hotel.nume%TYPE;
    BEGIN
        c := hoteluri_pachet_ex14.hoteluri_cu_modificari(0);
        IF c IS NOT NULL THEN
            LOOP
                FETCH c INTO cod_hotel_curent, nume_hotel_curent;
                EXIT WHEN c%NOTFOUND;
                DBMS_OUTPUT.PUT_LINE(nume_hotel_curent);
            END LOOP;
        END IF;
    END;
END;

DBMS_OUTPUT.PUT_LINE('*****');

BEGIN
    BEGIN
        hoteluri_pachet_ex14.marire_salarii_angajati(9);
        ROLLBACK;
    END;

    BEGIN
        UPDATE angajat
        SET salariu = 99999
        WHERE cod_hotel = 9 AND cod_manager IS NOT NULL;

        hoteluri_pachet_ex14.marire_salarii_angajati(9);
        ROLLBACK;
    END;
END;

DBMS_OUTPUT.PUT_LINE('*****');

BEGIN
    hoteluri_pachet_ex14.scadere_preturi_pachete(9);
    ROLLBACK;
END;

```

```

DBMS_OUTPUT.PUT_LINE('*****');

BEGIN
    BEGIN
        hoteluri_pachet_ex14.bonus_manageri;
        ROLLBACK;
    END;

    BEGIN
        UPDATE angajat
        SET salariu = 1000
        WHERE cod_manager IS NOT NULL;

        hoteluri_pachet_ex14.bonus_manageri;
        ROLLBACK;
    END;
END;

DBMS_OUTPUT.PUT_LINE('*****');

BEGIN
    -- marire salarii angajati
    BEGIN
        hoteluri_pachet_ex14.verificare_trimestriala;
        ROLLBACK;
    END;

    DBMS_OUTPUT.PUT_LINE('-----');

    -- scadere preturi pachete
    BEGIN
        INSERT INTO rezervare VALUES (19, 106, 149,
TO_DATE('01-01-2020', 'DD-MM-YYYY'), 99999);

        hoteluri_pachet_ex14.verificare_trimestriala;
        ROLLBACK;
    END;

    DBMS_OUTPUT.PUT_LINE('-----');

    -- bonus manageri
    BEGIN

```

```
        INSERT INTO rezervare VALUES (19, 106, 149,
TO_DATE('01-01-2020', 'DD-MM-YYYY'), 10720-1800);

        hoteluri_pachet_ex14.verificare_trimestriala;
        ROLLBACK;
END;

DBMS_OUTPUT.PUT_LINE( '-----');

-- exc_perioada
BEGIN
    DELETE FROM rezervare
    WHERE EXTRACT(YEAR FROM data_sosire) = 2021
        AND EXTRACT(MONTH FROM data_sosire) = 1 OR EXTRACT(MONTH FROM
data_sosire) = 2 OR EXTRACT(MONTH FROM data_sosire) = 3;

    hoteluri_pachet_ex14.verificare_trimestriala;
    ROLLBACK;
END;
END;
/
```

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          > COD_ANGAJAT
          > COD_HOTEL
          > NUME
          > VARD

```

Services

```

Tx | Output C##PROJECT_SGBD.ANGAJAT
  | [2024-01-11 08:45:31] completed in 125 ms
  | Trimestrul curent este: 1
  | ****
  | Ultimii ani inregistrati sunt: 2021 si 2020
  | ****
  | Incasările din trimestrul 1 al anului 2020 sunt: 1800
  | Nu există rezervări în trimestrul 3 al anului 2027
  | ****
  | Enache Corina 3900
  | Dumitrescu Raluca 4100
  | Stanescu Ana 4000
  | Cristea Dumitru 4000
  | Nu există angajați eligibili pentru marire
  | ****
  | Alpin
  | Vega
  | Novotel

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Cold weather

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          > COD_ANGAJAT
          > COD_HOTEL
          > NUME
          > VARD

```

Services

```

Tx | Output C##PROJECT_SGBD.ANGAJAT
  | Alpin
  | Vega
  | Novotel
  | Ramada Plaza
  | Riviera
  | Alpin
  | Tipul cauterării trebuie să fie -1 sau 1
  | ****
  | Angajatul Enache Corina a primit marire de salariu, noul salariu fiind 4400
  | Angajatul Dumitrescu Raluca a primit marire de salariu, noul salariu fiind 4600
  | Angajatul Stanescu Ana a primit marire de salariu, noul salariu fiind 4500
  | Angajatul Cristea Dumitru a primit marire de salariu, noul salariu fiind 4500
  | Nu există angajați eligibili pentru marire
  | ****
  | Pachetul cu beneficiul All Inclusive a scăzut pretul la 420
  | Pachetul cu beneficiul Drumatii pe munte a scăzut pretul la 320
  | Pachetul cu beneficiul Acces la Spa și Sala de fitness a scăzut pretul la 245
  | Pachetul cu beneficiul Acces la Sala de jocuri a scăzut pretul la 90

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Cold weather

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          > COD_ANGA...
          > COD_HOTEL

```

Services

Tx	Output
< Database	Pachetul cu beneficiul Urumetii pe munte a scazut pretul la 320
< project_sgbd	Pachetul cu beneficiul Acces la Spa si Sala de fitness a scazut pretul la 245
< default 10 ms	Pachetul cu beneficiul Acces la Sala de jocuri a scazut pretul la 90
< console_4	*****
< console_5	Managerul Constantinescu Diana a primit bonus, nou salaruu fiind 35610
< console_2	Managerul Ionescu Radu a primit bonus, nou salaruu fiind 33478
< console	Managerul Popescu Alexandru a primit bonus, nou salaruu fiind 32575
< console_8 184 ms	Managerul Marinescu Andrei a primit bonus, nou salaruu fiind 18340
< console_7	Managerul Valeriu George a primit bonus, nou salaruu fiind 22380
< console_6	Managerul Radu Mihai a primit bonus, nou salaruu fiind 27390
< console_3	Nu exista manageri eligibili pentru bonus
< console_1	*****
< console_7	Ultimii ani inregistrati pe care verificam performanta sunt: 2021 si 2020
< console_6	Incasarile curente sunt mai mari decat cele din anul precedent
< console_3	In hotelul Alpin:
< vlad	Angajatul Enache Corina a primit marire de salaruu, nou salaruu fiind 4400
< console_1	Angajatul Dumitrescu Raluca a primit marire de salaruu, nou salaruu fiind 4600
< console_1	Angajatul Stanescu Ana a primit marire de salaruu, nou salaruu fiind 4500

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Cold weather

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          > COD_ANGA...
          > COD_HOTEL

```

Services

Tx	Output
< Database	Angajatul Enache Corina a primit marire de salaruu, nou salaruu fiind 4400
< project_sgbd	Angajatul Dumitrescu Raluca a primit marire de salaruu, nou salaruu fiind 4600
< default 10 ms	Angajatul Stanescu Ana a primit marire de salaruu, nou salaruu fiind 4500
< console_4	Angajatul Cristea Dumitru a primit marire de salaruu, nou salaruu fiind 4500
< console_5	In hotelul Vega:
< console_2	Angajatul Nistor David a primit marire de salaruu, nou salaruu fiind 7000
< console	Angajatul Calinescu Cosmin a primit marire de salaruu, nou salaruu fiind 6600
< console_8 184 ms	Angajatul Cojocaru Cosmina a primit marire de salaruu, nou salaruu fiind 4450
< console_7	Angajatul Neagu Dana a primit marire de salaruu, nou salaruu fiind 4475
< console_6	Angajatul Ionita Irina a primit marire de salaruu, nou salaruu fiind 4400
< console_3	Angajatul Munteanu Mihai a primit marire de salaruu, nou salaruu fiind 5700
< vlad	Angajatul Toma Andrei a primit marire de salaruu, nou salaruu fiind 5280
< console_1	Angajatul Stanciu Stefan a primit marire de salaruu, nou salaruu fiind 5400
< console_7	-----
< console_6	Ultimii ani inregistrati pe care verificam performanta sunt: 2021 si 2020
< console_3	Incasarile curente sunt mai mici decat cele din anul precedent
< vlad	In hotelul Novotel:
< console_1	Pachetul cu beneficiul All Inclusive a scazut pretul la 470

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Cold weather

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          > COD_ANGAJAT
          > COD_HOTEL
  > NUME VARC

```

Services

Tx	Output
Database	C##PROJECT_SGBD.ANGAJAT
project_sgbd	Incasarile curente sunt mai mici decat cele din anul precedent
default 10 ms	Pachetul cu beneficiul All Inclusive a scazut pretul la 470
console_4	Pachetul cu beneficiul Acces la Spa si Sala de fitness a scazut pretul la 320
console_5	Pachetul cu beneficiul Acces la Piscina a scazut pretul la 200
console_2	Pachetul cu beneficiul Acces la muzeele din oras a scazut pretul la 170
console	Pachetul cu beneficiul Tur ghidat a scazut pretul la 140
console 184 ms	In hotelul Novotel:
console_7	Pachetul cu beneficiul All Inclusive a scazut pretul la 370
console_6	Pachetul cu beneficiul Acces la muzeele din oras a scazut pretul la 110
console_3	Pachetul cu beneficiul Acces la Piscina a scazut pretul la 120
vlad	In hotelul Riviera:
console_1	Pachetul cu beneficiul All Inclusive a scazut pretul la 370
console_7	Pachetul cu beneficiul Acces la Piscina a scazut pretul la 150
console_6	In hotelul Alpin:
console_3	Pachetul cu beneficiul All Inclusive a scazut pretul la 420
vlad	Pachetul cu beneficiul Drumetii pe munte a scazut pretul la 320
console_1	Pachetul cu beneficiul Acces la Spa si Sala de fitness a scazut pretul la 245

Database Consoles > project_sgbd > console_8 [project_sgbd]

502:5 CRLF UTF-8 4 spaces ⌂ ENG 1246 AM 1/11/2024

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          > COD_ANGAJAT
          > COD_HOTEL
  > NUME VARC

```

Services

Tx	Output
Database	C##PROJECT_SGBD.ANGAJAT
project_sgbd	Pachetul cu beneficiul ALL inclusive a scazut pretul la 420
default 10 ms	Pachetul cu beneficiul Drumetii pe munte a scazut pretul la 320
console_4	Pachetul cu beneficiul Acces la Spa si Sala de fitness a scazut pretul la 245
console_5	Pachetul cu beneficiul Acces la Sala de jocuri a scazut pretul la 90
console_2	-----
console	Ultimii ani inregistrati pe care verificam performanta sunt: 2021 si 2020
console 184 ms	Incasarile curente sunt egale cu cele din anul precedent
console_7	Managerul Constantinescu Diana a primit bonus, noul salariu fiind 35610
console_6	Managerul Ionescu Radu a primit bonus, noul salariu fiind 33478
console_3	Managerul Popescu Alexandru a primit bonus, noul salariu fiind 32575
vlad	Managerul Marinescu Andrei a primit bonus, noul salariu fiind 18340
console_1	Managerul Valeriu George a primit bonus, noul salariu fiind 22380
console_7	Managerul Radu Mihai a primit bonus, noul salariu fiind 27390
console_6	-----
console_3	Ultimii ani inregistrati pe care verificam performanta sunt: 2021 si 2020
vlad	Nu exista rezervari perioada curenta

Database Consoles > project_sgbd > console_8 [project_sgbd]

502:5 CRLF UTF-8 4 spaces ⌂ ENG 1246 AM 1/11/2024

Database Explorer

```

/*
 * In fiecare trimestru, se stabileste performanta hotelurilor fata de anul precedent, in aceeasi perioada, prin calculare a celor mai mari incasarilor totale avute. Sa se verifice daca incasarile totale pe ultimul an in care s-au inregistrat rezervari sunt mai mari decat incasarile totale pe anul precedent, raportat la trimestrul in care ne aflam in prezent (fiecare trimestru fiind codificat prin 1, 2, 3, in ordinea acestora din an).
 * In caz afirmativ, in cadrul hotelurilor in care s-au facut cel putin 3 rezervari cat este media numarului de rezervari per hotel (in cazul in care media nu va fi un numar intreg, se va lua partea intreaga a acestia), sa se mearasca cu 500 de lei salariul fiecarui angajat care era platit mai putin de 20% din salariul managerului sau. De asemenea, sa se afiseze numele angajatilor care au primit marirea de salariu, precum si nouul salariu al acestora.
 * In caz negativ, in care am avut mai multe rezervari cat media calculata anterior. De asemenea, sa se afiseze numele hotelurilor care au scuzat pretul pachetelor, precum si nouul pret al acestora.
 * In cazul in care incasarile sunt egale (performanta este aceeasi), fiecare manager care are marcaru un angajat cu salariu mai mare de 6000 de lei, sa primeasca un bonus la salariu egal cu 10% din salariul angajatului pe care il are in subordine cu cel mai mic salariu. De asemenea, sa se afiseze numele managerilor care au primit bonus, precum si nouul salariu al acestora.
 */
CREATE OR REPLACE PACKAGE hoteluri_pachet_ex14 IS
    FUNCTION trimestru_current RETURN NUMBER;
    PROCEDURE cei_mai_recenti_anii_inregistrati(an1 OUT NUMBER, an2 OUT NUMBER);

    TYPE t_lini IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
    TYPE t_mat IS TABLE OF t_lini INDEX BY PLS_INTEGER;
    mat_incarsi_t_mat := t_mat();
    PROCEDURE init_incarsi_trimestriale(an NUMBER, trimestru NUMBER);

    TYPE info_ang IS RECORD(
        cod_angajat angajat.cod_angajat%TYPE,
        nume_complet VARCHAR2(30),
        salariu_angajat.salariu%TYPE
    );

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Partly cloudy

Database Explorer

```

TYPE info_ang IS RECORD(
    cod_angajat angajat.cod_angajat%TYPE,
    nume_complet VARCHAR2(30),
    salariu_angajat.salariu%TYPE
);
TYPE tab_info_ang IS TABLE OF info_ang;
FUNCTION lista_angajati_eligibili_pentru_marire(cod_hotel_dorit hotel.cod_hotel%TYPE) RETURN tab_info_ang;

FUNCTION hoteluri_cu_modificari(tip_cautare NUMBER) RETURN SYS_REFCURSOR;

PROCEDURE marire_salarii_angajati(cod_hotel_dorit hotel.cod_hotel%TYPE);
PROCEDURE scadere_preturi_pachete(cod_hotel_dorit hotel.cod_hotel%TYPE);
PROCEDURE bonus_manageri;

PROCEDURE verificare_trimestriala;
END hoteluri_pachet_ex14;

CREATE OR REPLACE PACKAGE BODY hoteluri_pachet_ex14 IS
    FUNCTION trimestru_current RETURN NUMBER IS
        luna_curenta NUMBER;
    BEGIN
        SELECT EXTRACT(MONTH FROM SYSDATE) INTO luna_curenta
        FROM DUAL;
        CASE
            WHEN luna_curenta BETWEEN 1 AND 4 THEN
                RETURN 1;

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Partly cloudy

Database Explorer

```

55      WHEN luna_curenta BETWEEN 1 AND 4 THEN
56          RETURN 1;
57      WHEN luna_curenta BETWEEN 5 AND 8 THEN
58          RETURN 2;
59      WHEN luna_curenta BETWEEN 9 AND 12 THEN
60          RETURN 3;
61  END CASE;
62  -- calcul direct
63  --RETURN (luna_curenta - 1) / 4 + 1;
64
65  END trimestru_curent;

66 PROCEDURE cei_mai_recenti_anii_inregistrati(an1 OUT NUMBER, an2 OUT NUMBER) IS
67 BEGIN
68     -- stim deja ca lantul hotelier exista de mai mult de 2 ani
69     SELECT MAX(EXTRACT(YEAR FROM data_sosire)) INTO an1
70     FROM rezervare
71     FETCH FIRST ROW ONLY;
72
73     SELECT MAX(EXTRACT(YEAR FROM data_sosire)) INTO an2
74     FROM rezervare
75     WHERE EXTRACT(YEAR FROM data_sosire) <> an1
76     FETCH FIRST ROW ONLY;
77
78  END cei_mai_recenti_anii_inregistrati;

79 PROCEDURE init_incasari_trimestriale(an NUMBER, trimestru NUMBER) IS
80 BEGIN
81     SELECT SUM(pret) INTO mat_incasari(an)(trimestru)
82     FROM rezervare
83     WHERE EXTRACT(YEAR FROM data_sosire) = an
84
85     FROM rezervare
86     WHERE EXTRACT(YEAR FROM data_sosire) = an
87         AND (EXTRACT(MONTH FROM data_sosire) BETWEEN (trimestru - 1) * 4 + 1 AND trimestru * 4);
88
89     IF mat_incasari(an)(trimestru) IS NULL THEN
90         RAISE NO_DATA_FOUND;
91     END IF;

92     EXCEPTION
93         WHEN NO_DATA_FOUND THEN
94             mat_incasari(an)(trimestru) := -1;
95             DBMS_OUTPUT.PUT_LINE('Nu exista rezervari in trimestrul ' || trimestru || ' al anului ' || an);
96
97     END init_incasari_trimestriale;

98 FUNCTION lista_angajati_eligibili_pentru_marire(cod_hotel_dorit hotel.cod_hotel%TYPE) RETURN tab_info_ang IS
99     lista_info_ang tab_info_ang;
100    BEGIN
101        SELECT a.cod_angajat, a.nume || ' ' || a.prenume, a.salariu BULK COLLECT INTO lista_info_ang
102        FROM angajat a JOIN angajat man ON (a.cod_manager = man.cod_angajat)
103        WHERE a.cod_manager IS NOT NULL AND a.cod_hotel = cod_hotel_dorit
104            AND a.salariu < man.salariu * 0.2;

105        IF lista_info_ang.COUNT = 0 THEN
106            RAISE NO_DATA_FOUND;
107        END IF;

108        RETURN lista_info_ang;
109
110    EXCEPTION
111

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Partly cloudy

Database Explorer

```

82
83     FROM rezervare
84     WHERE EXTRACT(YEAR FROM data_sosire) = an
85         AND (EXTRACT(MONTH FROM data_sosire) BETWEEN (trimestru - 1) * 4 + 1 AND trimestru * 4);

86     IF mat_incasari(an)(trimestru) IS NULL THEN
87         RAISE NO_DATA_FOUND;
88     END IF;

89     EXCEPTION
90         WHEN NO_DATA_FOUND THEN
91             mat_incasari(an)(trimestru) := -1;
92             DBMS_OUTPUT.PUT_LINE('Nu exista rezervari in trimestrul ' || trimestru || ' al anului ' || an);
93
94     END init_incasari_trimestriale;

95 FUNCTION lista_angajati_eligibili_pentru_marire(cod_hotel_dorit hotel.cod_hotel%TYPE) RETURN tab_info_ang IS
96     lista_info_ang tab_info_ang;
97    BEGIN
98        SELECT a.cod_angajat, a.nume || ' ' || a.prenume, a.salariu BULK COLLECT INTO lista_info_ang
99        FROM angajat a JOIN angajat man ON (a.cod_manager = man.cod_angajat)
100        WHERE a.cod_manager IS NOT NULL AND a.cod_hotel = cod_hotel_dorit
101            AND a.salariu < man.salariu * 0.2;

102        IF lista_info_ang.COUNT = 0 THEN
103            RAISE NO_DATA_FOUND;
104        END IF;

105        RETURN lista_info_ang;
106
107    EXCEPTION
108

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Partly cloudy

Database Explorer

```

111 WHEN NO_DATA_FOUND THEN
112   DBMS_OUTPUT.PUT_LINE( A_>'Nu există angajați eligibili pentru marire');
113   RETURN NULL;
114
115 END lista_angajati_eligibili_pentru_marire;
116
117 FUNCTION hoteluri_cu_modificari(tip_cautare NUMBER) RETURN SYS_REFCURSOR IS
118   c SYS_REFCURSOR;
119   nr_mediul_rezervari NUMBER;
120   exc_tip_cautare EXCEPTION;
121   PRAGMA EXCEPTION_INIT(exc_tip_cautare, -20001);
122
123   BEGIN
124     IF tip_cautare <> -1 AND tip_cautare <> 1 THEN
125       RAISE exc_tip_cautare;
126     END IF;
127
128     SELECT FLOOR(AVG(COUNT(*))) INTO nr_mediul_rezervari
129     FROM rezervare r JOIN camera cam ON (r.cod_camera = cam.cod_camera)
130     GROUP BY cam.cod_hotel;
131
132     -- SQL dinamic: tip_cautare = 1 => creează prețul, tip_cautare = -1 => scade prețul
133     -- sigur avem hoteluri eligibile pentru modificări (meru există macar un hotel sub medie și unul peste medie)
134     OPEN c FOR
135       'SELECT h.cod_hotel, h.nume ' ||
136       'FROM rezervare r JOIN camera cam ON (r.cod_camera = cam.cod_camera) ' ||
137       'JOIN hotel h ON (cam.cod_hotel = h.cod_hotel) ' ||
138       'GROUP BY h.cod_hotel, h.nume ' ||
139       'HAVING (COUNT(*) - :nr_mediul_rezervari) * :tip_cautare >= 0'
140       USING nr_mediul_rezervari, tip_cautare;
141
142   EXCEPTION
143     WHEN exc_tip_cautare THEN
144       DBMS_OUTPUT.PUT_LINE( A_>'Tipul căutării trebuie să fie -1 sau 1');
145       RETURN NULL;
146   END hoteluri_cu_modificari;
147
148
149 PROCEDURE marire_salarii_angajati(cod_hotel_dorit hotel.cod_hotel%TYPE) IS
150   lista_info_ang tab_info_ang;
151
152   BEGIN
153     lista_info_ang := lista_angajati_eligibili_pentru_marire( cod_hotel_dorit: cod_hotel_dorit );
154     IF lista_info_ang IS NOT NULL THEN
155       FOR i IN 1 .. lista_info_ang.COUNT LOOP
156         UPDATE angajat a
157           SET a.salariu = a.salariu + 500
158           WHERE a.cod_angajat = lista_info_ang(i).cod_angajat;
159         lista_info_ang(i).salariu := lista_info_ang(i).salariu + 500;
160         DBMS_OUTPUT.PUT_LINE( A_>'Angajatul ' || lista_info_ang(i).nume_complet || ' a primit marire de salariu, noul salariu fiind ');
161       END LOOP;
162     END IF;
163     END marire_salarii_angajati;
164
165
166 PROCEDURE scadere_preturi_pachete(cod_hotel_dorit hotel.cod_hotel%TYPE) IS
167   -- fiecare hotel oferă cel puțin un pachet
168   CURSOR pachetele_hotelului IS
169

```

Database Consoles > project_sgbdb > console_8 [project_sgbdb]

-5°C Partly cloudy

498:58 CRLF UTF-8 4 spaces ENG 12:23 AM 1/11/2024

Database Explorer

```

138   'HAVING (COUNT(*) - :nr_mediul_rezervari) * :tip_cautare >= 0'
139   USING nr_mediul_rezervari, tip_cautare;
140
141   RETURN c;
142
143 EXCEPTION
144   WHEN exc_tip_cautare THEN
145     DBMS_OUTPUT.PUT_LINE( A_>'Tipul căutării trebuie să fie -1 sau 1');
146     RETURN NULL;
147 END hoteluri_cu_modificari;
148
149
150 PROCEDURE marire_salarii_angajati(cod_hotel_dorit hotel.cod_hotel%TYPE) IS
151   lista_info_ang tab_info_ang;
152
153   BEGIN
154     lista_info_ang := lista_angajati_eligibili_pentru_marire( cod_hotel_dorit: cod_hotel_dorit );
155     IF lista_info_ang IS NOT NULL THEN
156       FOR i IN 1 .. lista_info_ang.COUNT LOOP
157         UPDATE angajat a
158           SET a.salariu = a.salariu + 500
159           WHERE a.cod_angajat = lista_info_ang(i).cod_angajat;
160         lista_info_ang(i).salariu := lista_info_ang(i).salariu + 500;
161         DBMS_OUTPUT.PUT_LINE( A_>'Angajatul ' || lista_info_ang(i).nume_complet || ' a primit marire de salariu, noul salariu fiind ');
162       END LOOP;
163     END IF;
164     END marire_salarii_angajati;
165
166
167 PROCEDURE scadere_preturi_pachete(cod_hotel_dorit hotel.cod_hotel%TYPE) IS
168   -- fiecare hotel oferă cel puțin un pachet
169   CURSOR pachetele_hotelului IS
170

```

Database Consoles > project_sgbdb > console_8 [project_sgbdb]

-5°C Partly cloudy

498:58 CRLF UTF-8 4 spaces ENG 12:23 AM 1/11/2024

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- Title Bar:** Shows "sqlproject" and "Version control".
- Database Explorer:** On the left, it lists databases "project_2 of 40" and "project_sgbdb 2 of 41". Under "project_sgbdb", there are tables like ANEXA, ANGAJAT, and PACHET, along with columns and keys.
- SQL Editor:** The main area displays a PL/SQL script. The code includes:
 - A cursor loop that updates hotel prices by 30 units.
 - A procedure "bonus_manageri" that defines a record type for manager information and a table of managers.
- Status Bar:** Shows file statistics: 11 lines and 344 characters.
- Bottom Bar:** Includes icons for search, file operations, and extensions like Prettier and ESLint.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Shows "sqlproject" and "Version control".
- Database Explorer:** Shows a tree view of the database schema for "project_sgbd". The "ANGAJAT" table is expanded, showing columns: COD_ANGAJAT, COD_HOTEL, NUME, PRENUME, DATA_NASTRIERI, JOB, SALARIU, and COD_MANAGER.
- Code Editor:** Displays a PL/SQL script named "hoteli_cu_modificari". The code implements a stored procedure or function that finds the minimum salary for managers whose manager is null. It then iterates through all employees, comparing their salary to the minimum found. If an employee's salary is less than the current minimum, it updates the minimum. Finally, it checks if any manager has a salary greater than 6000 and updates their manager's eligible status accordingly.
- Taskbar:** Shows file paths like "C##PROJECT_SGBD" and "console_8 [project_sgbd]". Status icons include "11 344" notifications, "498:58 CRLF UTF-8", and "4 spaces".

Database Explorer

```

222 IF manageri_eligibili.COUNT = 0 THEN
223     RAISE exc_manageri;
224 END IF;

225 FOR i IN 1..manageri_eligibili.COUNT LOOP
226     UPDATE angajat
227     SET salariu = salariu + 0.1 * manageri_eligibili(i).salariu_minim
228     WHERE cod_angajat = manageri_eligibili(i).cod_angajat;
229     manageri_eligibili(i).salariu := manageri_eligibili(i).salariu + 0.1 * manageri_eligibili(i).salariu_minim;
230     DBMS_OUTPUT.PUT_LINE( A: 'Managerul ' || manageri_eligibili(i).nume_complet || ' a primit bonus, nou salariu fiind ' || manageri_eligibili(i).salariu );
231 END LOOP;

232 EXCEPTION
233     WHEN exc_manageri THEN
234         DBMS_OUTPUT.PUT_LINE( A: 'Nu exista manageri eligibili pentru bonus' );

235 END bonus_manageri;

236 PROCEDURE verificare_trimestriale IS
237     an1 NUMBER; an2 NUMBER;
238     cod_hotel_current hotel.cod_hotel%TYPE;
239     nume_hotel_current hotel.nume%TYPE;
240     c SYS_REFCURSOR;

241     exc_perioada EXCEPTION;
242     PRAGMA EXCEPTION_INIT(exc_perioada, -20001);

243 BEGIN
244     cei_mai_recenti_anii_inregistrati( an1: an1, an2: an2);
245     DBMS_OUTPUT.PUT_LINE( A: 'Ultimi anii inregistrati pe care verificam performanta sunt: ' || an1 || ' si ' || an2);
246
247     init_incasari_trimestriale( an: an1, trimestru: trimestru_current);
248     init_incasari_trimestriale( an: an2, trimestru: trimestru_current);

249     -- daca nu exista rezervari in trimestrul trecut, punand -1 in matrice, vom avea incasările curente mai mari
250     IF mat_incasari(an1)(trimestru_current) = -1 THEN
251         RAISE exc_perioada;
252     END IF;

253     CASE
254         WHEN mat_incasari(an1)(trimestru_current) > mat_incasari(an2)(trimestru_current) THEN
255             DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt mai mari decat cele din anul precedent' );
256             c := hoteluri_cu_modificari( tip_cautare: 1 );
257
258             LOOP
259                 FETCH c INTO cod_hotel_current, nume_hotel_current;
260                 EXIT WHEN c%NOTFOUND;
261                 DBMS_OUTPUT.PUT_LINE( A: 'In hotelul ' || ' ' || nume_hotel_current || ' : ' );
262                 marire_salariu_angajati( cod_hotel_dorit: cod_hotel_current );
263             END LOOP;

264         WHEN mat_incasari(an1)(trimestru_current) < mat_incasari(an2)(trimestru_current) THEN
265             DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt mai mici decat cele din anul precedent' );
266             c := hoteluri_cu_modificari( tip_cautare: -1 );
267
268             LOOP
269                 FETCH c INTO cod_hotel_current, nume_hotel_current;
270                 EXIT WHEN c%NOTFOUND;
271                 DBMS_OUTPUT.PUT_LINE( A: 'In hotelul ' || ' ' || nume_hotel_current || ' : ' );
272                 marire_salariu_angajati( cod_hotel_dorit: cod_hotel_current );
273             END LOOP;
274
275     END CASE;
276
277     DBMS_OUTPUT.PUT_LINE( A: 'Fisier salvat cu numele: ' || an1 || '-' || an2 || '.txt' );
278
279 END;

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Party cloudy 498:58 CRLF UTF-8 4 spaces ENG 12:23 AM 1/11/2024

Database Explorer

```

249 BEGIN
250     cei_mai_recenti_anii_inregistrati( an1: an1, an2: an2);
251     DBMS_OUTPUT.PUT_LINE( A: 'Ultimi anii inregistrati pe care verificam performanta sunt: ' || an1 || ' si ' || an2);

252     init_incasari_trimestriale( an: an1, trimestru: trimestru_current);
253     init_incasari_trimestriale( an: an2, trimestru: trimestru_current);

254     -- daca nu exista rezervari in trimestrul trecut, punand -1 in matrice, vom avea incasările curente mai mari
255     IF mat_incasari(an1)(trimestru_current) = -1 THEN
256         RAISE exc_perioada;
257     END IF;

258     CASE
259         WHEN mat_incasari(an1)(trimestru_current) > mat_incasari(an2)(trimestru_current) THEN
260             DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt mai mari decat cele din anul precedent' );
261             c := hoteluri_cu_modificari( tip_cautare: 1 );
262
263             LOOP
264                 FETCH c INTO cod_hotel_current, nume_hotel_current;
265                 EXIT WHEN c%NOTFOUND;
266                 DBMS_OUTPUT.PUT_LINE( A: 'In hotelul ' || ' ' || nume_hotel_current || ' : ' );
267                 marire_salariu_angajati( cod_hotel_dorit: cod_hotel_current );
268             END LOOP;

269         WHEN mat_incasari(an1)(trimestru_current) < mat_incasari(an2)(trimestru_current) THEN
270             DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt mai mici decat cele din anul precedent' );
271             c := hoteluri_cu_modificari( tip_cautare: -1 );
272
273             LOOP
274                 FETCH c INTO cod_hotel_current, nume_hotel_current;
275                 EXIT WHEN c%NOTFOUND;
276                 DBMS_OUTPUT.PUT_LINE( A: 'In hotelul ' || ' ' || nume_hotel_current || ' : ' );
277                 marire_salariu_angajati( cod_hotel_dorit: cod_hotel_current );
278             END LOOP;
279
280     END CASE;
281
282     DBMS_OUTPUT.PUT_LINE( A: 'Fisier salvat cu numele: ' || an1 || '-' || an2 || '.txt' );
283
284 END;

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Party cloudy 498:58 CRLF UTF-8 4 spaces ENG 12:24 AM 1/11/2024

Database Explorer

```

    WHEN mat_incasari(an1)(trimestru_current) < mat_incasari(an2)(trimestru_current) THEN
        DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt mai mici decat cele din anul precedent');
        c := hoteluri_cu_modificari( tip_cautare: -1);

        LOOP
            FETCH c INTO cod_hotel_current, nume_hotel_current;
            EXIT WHEN c%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE( A: 'In hotelul' || ' ' || nume_hotel_current || ': ');
            scadere_preturi_pachete( cod_hotel_dorit; cod_hotel_current);
        END LOOP;

        WHEN mat_incasari(an1)(trimestru_current) = mat_incasari(an2)(trimestru_current) THEN
            DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt egale cu cele din anul precedent');
            bonus_manageri;
        END CASE;

        EXCEPTION
            WHEN exc_perioada THEN
                DBMS_OUTPUT.PUT_LINE( A: 'Nu exista rezervari perioada curenta');
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE( A: 'Alta eroare: ' || SQLERRM);
        END verificare_trimestriala;

    END hoteluri_pachet_ex14;
/
BEGIN
    BEGIN
        DBMS_OUTPUT.PUT_LINE( A: 'Trimestrul curent este: ' || hoteluri_pachet_ex14.trimestru_current);
    END;

    DBMS_OUTPUT.PUT_LINE( A: '*****');

    DECLARE
        an1 NUMBER; an2 NUMBER;
    BEGIN
        hoteluri_pachet_ex14.cei_mai_recenti_anii_inregistrati( AN1: an1, AN2: an2);
        DBMS_OUTPUT.PUT_LINE( A: 'Ultimii ani inregistrati sunt: ' || an1 || ' si ' || an2);
    END;

    DBMS_OUTPUT.PUT_LINE( A: '*****');

    DECLARE
        an1 NUMBER := 2020;
        trimestru1 NUMBER := 1;
        an2 NUMBER := 2027;
        trimestru2 NUMBER := 3;
    BEGIN
        hoteluri_pachet_ex14.init_incasari_trimestriale( AN: an1, TRIMESTRU: trimestru1);
        IF hoteluri_pachet_ex14.mat_incasari(an1)(trimestru1) <> -1 THEN
            DBMS_OUTPUT.PUT_LINE( A: 'Incasarile din trimestrul ' || trimestru1 || ' al anului ' || an1 || ' sunt: ' || hoteluri_pachet_ex14);
        ELSE
            DBMS_OUTPUT.PUT_LINE( A: 'Nu exista rezervari in trimestrul ' || trimestru1 || ' al anului ' || an1);
        END IF;
    END;

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

498:58 CRLF UTF-8 4 spaces ENG 12:24 AM 1/11/2024

Database Explorer

```

    WHEN mat_incasari(an1)(trimestru_current) < mat_incasari(an2)(trimestru_current) THEN
        DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt mai mici decat cele din anul precedent');
        c := hoteluri_cu_modificari( tip_cautare: -1);

        LOOP
            FETCH c INTO cod_hotel_current, nume_hotel_current;
            EXIT WHEN c%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE( A: 'In hotelul' || ' ' || nume_hotel_current || ': ');
            scadere_preturi_pachete( cod_hotel_dorit; cod_hotel_current);
        END LOOP;

        WHEN mat_incasari(an1)(trimestru_current) = mat_incasari(an2)(trimestru_current) THEN
            DBMS_OUTPUT.PUT_LINE( A: 'Incasarile curente sunt egale cu cele din anul precedent');
            bonus_manageri;
        END CASE;

        EXCEPTION
            WHEN exc_perioada THEN
                DBMS_OUTPUT.PUT_LINE( A: 'Nu exista rezervari perioada curenta');
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE( A: 'Alta eroare: ' || SQLERRM);
        END verificare_trimestriala;

    END hoteluri_pachet_ex14;
/
BEGIN
    BEGIN
        DBMS_OUTPUT.PUT_LINE( A: 'Trimestrul curent este: ' || hoteluri_pachet_ex14.trimestru_current);
    END;

    DBMS_OUTPUT.PUT_LINE( A: '*****');

    DECLARE
        an1 NUMBER; an2 NUMBER;
    BEGIN
        hoteluri_pachet_ex14.cei_mai_recenti_anii_inregistrati( AN1: an1, AN2: an2);
        DBMS_OUTPUT.PUT_LINE( A: 'Ultimii ani inregistrati sunt: ' || an1 || ' si ' || an2);
    END;

    DBMS_OUTPUT.PUT_LINE( A: '*****');

    DECLARE
        an1 NUMBER := 2020;
        trimestru1 NUMBER := 1;
        an2 NUMBER := 2027;
        trimestru2 NUMBER := 3;
    BEGIN
        hoteluri_pachet_ex14.init_incasari_trimestriale( AN: an1, TRIMESTRU: trimestru1);
        IF hoteluri_pachet_ex14.mat_incasari(an1)(trimestru1) <> -1 THEN
            DBMS_OUTPUT.PUT_LINE( A: 'Incasarile din trimestrul ' || trimestru1 || ' al anului ' || an1 || ' sunt: ' || hoteluri_pachet_ex14);
        ELSE
            DBMS_OUTPUT.PUT_LINE( A: 'Nu exista rezervari in trimestrul ' || trimestru1 || ' al anului ' || an1);
        END IF;
    END;

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

498:58 CRLF UTF-8 4 spaces ENG 12:24 AM 1/11/2024

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The top navigation bar includes 'sqlproject' and 'Version control'. The 'Database Explorer' sidebar lists databases like 'project_2 of 40' and 'project_sgbdb 2 of 41'. The 'Object Explorer' sidebar shows the schema of 'C##PROJECT_SGBDB', including tables such as 'ANEXA', 'ANGAJAT', 'DATA_NAST', 'JOB', 'SALARIU', 'COD_MANAG', 'keys', 'foreign keys', 'indexes', 'APROVIZIONARE', 'CAMERA', 'CLIENT', 'ERORI', 'FACILITATE', 'HOTEL', 'LOCATIE', 'OFERTA', 'PACHET', and 'COD_PACHE'. The main code editor window displays a PL/SQL script for a stored procedure named 'hoteluri_cu_modificari'. The script uses DBMS_OUTPUT.PUT_LINE to print messages and update the 'angajat' table by setting 'salariu' to 99999 for all rows.

```
327      END IF;
328
329      hoteluri_pachet_ex14.init_incasarri_trimestriale( AN.an2, TRIMESTRU: trimestru2);
330      IF hoteluri_pachet_ex14.mat_incasarri(an2)(trimestrul2) <> -1 THEN
331          DBMS_OUTPUT.PUT_LINE( A: 'Incasarile din trimestrul ' || trimestru2 || ' al anului ' || an2 || ' sunt: ' || hoteluri_pachet_ex14.mat_incasarri(an2)(trimestrul2));
332      ELSE
333          DBMS_OUTPUT.PUT_LINE( A: 'Nu exista rezervari in trimestrul ' || trimestru2 || ' al anului ' || an2);
334      END IF;
335
336      END;
337
338      DBMS_OUTPUT.PUT_LINE( A: '*****');
339
340
341      BEGIN
342          DECLARE
343              lista_ang hoteluri_pachet_ex14.tab_info_ang;
344          BEGIN
345              lista_ang := hoteluri_pachet_ex14.lista_angajati_eligibili_pentru_marire( COD_HOTEL_DORIT: 9);
346              IF lista_ang IS NOT NULL THEN
347                  FOR i IN 1..lista_ang.COUNT LOOP
348                      DBMS_OUTPUT.PUT_LINE( A: lista_ang(i).nume_complet || ' ' || lista_ang(i).salariu);
349                  END LOOP;
350              END IF;
351          END;
352
353          BEGIN
354              UPDATE angajat
355              SET salariu = 99999
356
357      END;
358
359      BEGIN
360          DECLARE
361              lista_ang hoteluri_pachet_ex14.tab_info_ang;
362          BEGIN
363              UPDATE angajat
364              SET salariu = 99999
365
366      END;
367
368      BEGIN
369          UPDATE angajat
370          SET salariu = 99999
371
372      END;
373
374      BEGIN
375          UPDATE angajat
376          SET salariu = 99999
377
378      END;
379
380      BEGIN
381          UPDATE angajat
382          SET salariu = 99999
383
384      END;
385
386      BEGIN
387          UPDATE angajat
388          SET salariu = 99999
389
390      END;
391
392      BEGIN
393          UPDATE angajat
394          SET salariu = 99999
395
396      END;
397
398      BEGIN
399          UPDATE angajat
400          SET salariu = 99999
401
402      END;
403
404      BEGIN
405          UPDATE angajat
406          SET salariu = 99999
407
408      END;
409
410      BEGIN
411          UPDATE angajat
412          SET salariu = 99999
413
414      END;
415
416      BEGIN
417          UPDATE angajat
418          SET salariu = 99999
419
420      END;
421
422      BEGIN
423          UPDATE angajat
424          SET salariu = 99999
425
426      END;
427
428      BEGIN
429          UPDATE angajat
430          SET salariu = 99999
431
432      END;
433
434      BEGIN
435          UPDATE angajat
436          SET salariu = 99999
437
438      END;
439
440      BEGIN
441          UPDATE angajat
442          SET salariu = 99999
443
444      END;
445
446      BEGIN
447          UPDATE angajat
448          SET salariu = 99999
449
450      END;
451
452      BEGIN
453          UPDATE angajat
454          SET salariu = 99999
455
456      END;
457
458      BEGIN
459          UPDATE angajat
460          SET salariu = 99999
461
462      END;
463
464      BEGIN
465          UPDATE angajat
466          SET salariu = 99999
467
468      END;
469
470      BEGIN
471          UPDATE angajat
472          SET salariu = 99999
473
474      END;
475
476      BEGIN
477          UPDATE angajat
478          SET salariu = 99999
479
480      END;
481
482      BEGIN
483          UPDATE angajat
484          SET salariu = 99999
485
486      END;
487
488      BEGIN
489          UPDATE angajat
490          SET salariu = 99999
491
492      END;
493
494      BEGIN
495          UPDATE angajat
496          SET salariu = 99999
497
498      END;
499
500      BEGIN
501          UPDATE angajat
502          SET salariu = 99999
503
504      END;
505
506      BEGIN
507          UPDATE angajat
508          SET salariu = 99999
509
510      END;
511
512      BEGIN
513          UPDATE angajat
514          SET salariu = 99999
515
516      END;
517
518      BEGIN
519          UPDATE angajat
520          SET salariu = 99999
521
522      END;
523
524      BEGIN
525          UPDATE angajat
526          SET salariu = 99999
527
528      END;
529
530      BEGIN
531          UPDATE angajat
532          SET salariu = 99999
533
534      END;
535
536      BEGIN
537          UPDATE angajat
538          SET salariu = 99999
539
540      END;
541
542      BEGIN
543          UPDATE angajat
544          SET salariu = 99999
545
546      END;
547
548      BEGIN
549          UPDATE angajat
550          SET salariu = 99999
551
552      END;
553
554      BEGIN
555          UPDATE angajat
556          SET salariu = 99999
557
558      END;
559
560      BEGIN
561          UPDATE angajat
562          SET salariu = 99999
563
564      END;
565
566      BEGIN
567          UPDATE angajat
568          SET salariu = 99999
569
570      END;
571
572      BEGIN
573          UPDATE angajat
574          SET salariu = 99999
575
576      END;
577
578      BEGIN
579          UPDATE angajat
580          SET salariu = 99999
581
582      END;
583
584      BEGIN
585          UPDATE angajat
586          SET salariu = 99999
587
588      END;
589
590      BEGIN
591          UPDATE angajat
592          SET salariu = 99999
593
594      END;
595
596      BEGIN
597          UPDATE angajat
598          SET salariu = 99999
599
600      END;
601
602      BEGIN
603          UPDATE angajat
604          SET salariu = 99999
605
606      END;
607
608      BEGIN
609          UPDATE angajat
610          SET salariu = 99999
611
612      END;
613
614      BEGIN
615          UPDATE angajat
616          SET salariu = 99999
617
618      END;
619
620      BEGIN
621          UPDATE angajat
622          SET salariu = 99999
623
624      END;
625
626      BEGIN
627          UPDATE angajat
628          SET salariu = 99999
629
630      END;
631
632      BEGIN
633          UPDATE angajat
634          SET salariu = 99999
635
636      END;
637
638      BEGIN
639          UPDATE angajat
640          SET salariu = 99999
641
642      END;
643
644      BEGIN
645          UPDATE angajat
646          SET salariu = 99999
647
648      END;
649
650      BEGIN
651          UPDATE angajat
652          SET salariu = 99999
653
654      END;
655
656      BEGIN
657          UPDATE angajat
658          SET salariu = 99999
659
660      END;
661
662      BEGIN
663          UPDATE angajat
664          SET salariu = 99999
665
666      END;
667
668      BEGIN
669          UPDATE angajat
670          SET salariu = 99999
671
672      END;
673
674      BEGIN
675          UPDATE angajat
676          SET salariu = 99999
677
678      END;
679
680      BEGIN
681          UPDATE angajat
682          SET salariu = 99999
683
684      END;
685
686      BEGIN
687          UPDATE angajat
688          SET salariu = 99999
689
690      END;
691
692      BEGIN
693          UPDATE angajat
694          SET salariu = 99999
695
696      END;
697
698      BEGIN
699          UPDATE angajat
700          SET salariu = 99999
701
702      END;
703
704      BEGIN
705          UPDATE angajat
706          SET salariu = 99999
707
708      END;
709
710      BEGIN
711          UPDATE angajat
712          SET salariu = 99999
713
714      END;
715
716      BEGIN
717          UPDATE angajat
718          SET salariu = 99999
719
720      END;
721
722      BEGIN
723          UPDATE angajat
724          SET salariu = 99999
725
726      END;
727
728      BEGIN
729          UPDATE angajat
730          SET salariu = 99999
731
732      END;
733
734      BEGIN
735          UPDATE angajat
736          SET salariu = 99999
737
738      END;
739
740      BEGIN
741          UPDATE angajat
742          SET salariu = 99999
743
744      END;
745
746      BEGIN
747          UPDATE angajat
748          SET salariu = 99999
749
750      END;
751
752      BEGIN
753          UPDATE angajat
754          SET salariu = 99999
755
756      END;
757
758      BEGIN
759          UPDATE angajat
760          SET salariu = 99999
761
762      END;
763
764      BEGIN
765          UPDATE angajat
766          SET salariu = 99999
767
768      END;
769
770      BEGIN
771          UPDATE angajat
772          SET salariu = 99999
773
774      END;
775
776      BEGIN
777          UPDATE angajat
778          SET salariu = 99999
779
780      END;
781
782      BEGIN
783          UPDATE angajat
784          SET salariu = 99999
785
786      END;
787
788      BEGIN
789          UPDATE angajat
790          SET salariu = 99999
791
792      END;
793
794      BEGIN
795          UPDATE angajat
796          SET salariu = 99999
797
798      END;
799
800      BEGIN
801          UPDATE angajat
802          SET salariu = 99999
803
804      END;
805
806      BEGIN
807          UPDATE angajat
808          SET salariu = 99999
809
810      END;
811
812      BEGIN
813          UPDATE angajat
814          SET salariu = 99999
815
816      END;
817
818      BEGIN
819          UPDATE angajat
820          SET salariu = 99999
821
822      END;
823
824      BEGIN
825          UPDATE angajat
826          SET salariu = 99999
827
828      END;
829
830      BEGIN
831          UPDATE angajat
832          SET salariu = 99999
833
834      END;
835
836      BEGIN
837          UPDATE angajat
838          SET salariu = 99999
839
840      END;
841
842      BEGIN
843          UPDATE angajat
844          SET salariu = 99999
845
846      END;
847
848      BEGIN
849          UPDATE angajat
850          SET salariu = 99999
851
852      END;
853
854      BEGIN
855          UPDATE angajat
856          SET salariu = 99999
857
858      END;
859
860      BEGIN
861          UPDATE angajat
862          SET salariu = 99999
863
864      END;
865
866      BEGIN
867          UPDATE angajat
868          SET salariu = 99999
869
870      END;
871
872      BEGIN
873          UPDATE angajat
874          SET salariu = 99999
875
876      END;
877
878      BEGIN
879          UPDATE angajat
880          SET salariu = 99999
881
882      END;
883
884      BEGIN
885          UPDATE angajat
886          SET salariu = 99999
887
888      END;
889
890      BEGIN
891          UPDATE angajat
892          SET salariu = 99999
893
894      END;
895
896      BEGIN
897          UPDATE angajat
898          SET salariu = 99999
899
900      END;
901
902      BEGIN
903          UPDATE angajat
904          SET salariu = 99999
905
906      END;
907
908      BEGIN
909          UPDATE angajat
910          SET salariu = 99999
911
912      END;
913
914      BEGIN
915          UPDATE angajat
916          SET salariu = 99999
917
918      END;
919
920      BEGIN
921          UPDATE angajat
922          SET salariu = 99999
923
924      END;
925
926      BEGIN
927          UPDATE angajat
928          SET salariu = 99999
929
930      END;
931
932      BEGIN
933          UPDATE angajat
934          SET salariu = 99999
935
936      END;
937
938      BEGIN
939          UPDATE angajat
940          SET salariu = 99999
941
942      END;
943
944      BEGIN
945          UPDATE angajat
946          SET salariu = 99999
947
948      END;
949
950      BEGIN
951          UPDATE angajat
952          SET salariu = 99999
953
954      END;
955
956      BEGIN
957          UPDATE angajat
958          SET salariu = 99999
959
960      END;
961
962      BEGIN
963          UPDATE angajat
964          SET salariu = 99999
965
966      END;
967
968      BEGIN
969          UPDATE angajat
970          SET salariu = 99999
971
972      END;
973
974      BEGIN
975          UPDATE angajat
976          SET salariu = 99999
977
978      END;
979
980      BEGIN
981          UPDATE angajat
982          SET salariu = 99999
983
984      END;
985
986      BEGIN
987          UPDATE angajat
988          SET salariu = 99999
989
990      END;
991
992      BEGIN
993          UPDATE angajat
994          SET salariu = 99999
995
996      END;
997
998      BEGIN
999          UPDATE angajat
1000          SET salariu = 99999
1001
1002      END;
1003
1004      BEGIN
1005          UPDATE angajat
1006          SET salariu = 99999
1007
1008      END;
1009
1010      BEGIN
1011          UPDATE angajat
1012          SET salariu = 99999
1013
1014      END;
1015
1016      BEGIN
1017          UPDATE angajat
1018          SET salariu = 99999
1019
1020      END;
1021
1022      BEGIN
1023          UPDATE angajat
1024          SET salariu = 99999
1025
1026      END;
1027
1028      BEGIN
1029          UPDATE angajat
1030          SET salariu = 99999
1031
1032      END;
1033
1034      BEGIN
1035          UPDATE angajat
1036          SET salariu = 99999
1037
1038      END;
1039
1040      BEGIN
1041          UPDATE angajat
1042          SET salariu = 99999
1043
1044      END;
1045
1046      BEGIN
1047          UPDATE angajat
1048          SET salariu = 99999
1049
1050      END;
1051
1052      BEGIN
1053          UPDATE angajat
1054          SET salariu = 99999
1055
1056      END;
1057
1058      BEGIN
1059          UPDATE angajat
1060          SET salariu = 99999
1061
1062      END;
1063
1064      BEGIN
1065          UPDATE angajat
1066          SET salariu = 99999
1067
1068      END;
1069
1070      BEGIN
1071          UPDATE angajat
1072          SET salariu = 99999
1073
1074      END;
1075
1076      BEGIN
1077          UPDATE angajat
1078          SET salariu = 99999
1079
1080      END;
1081
1082      BEGIN
1083          UPDATE angajat
1084          SET salariu = 99999
1085
1086      END;
1087
1088      BEGIN
1089          UPDATE angajat
1090          SET salariu = 99999
1091
1092      END;
1093
1094      BEGIN
1095          UPDATE angajat
1096          SET salariu = 99999
1097
1098      END;
1099
1100      BEGIN
1101          UPDATE angajat
1102          SET salariu = 99999
1103
1104      END;
1105
1106      BEGIN
1107          UPDATE angajat
1108          SET salariu = 99999
1109
1110      END;
1111
1112      BEGIN
1113          UPDATE angajat
1114          SET salariu = 99999
1115
1116      END;
1117
1118      BEGIN
1119          UPDATE angajat
1120          SET salariu = 99999
1121
1122      END;
1123
1124      BEGIN
1125          UPDATE angajat
1126          SET salariu = 99999
1127
1128      END;
1129
1130      BEGIN
1131          UPDATE angajat
1132          SET salariu = 99999
1133
1134      END;
1135
1136      BEGIN
1137          UPDATE angajat
1138          SET salariu = 99999
1139
1140      END;
1141
1142      BEGIN
1143          UPDATE angajat
1144          SET salariu = 99999
1145
1146      END;
1147
1148      BEGIN
1149          UPDATE angajat
1150          SET salariu = 99999
1151
1152      END;
1153
1154      BEGIN
1155          UPDATE angajat
1156          SET salariu = 99999
1157
1158      END;
1159
1160      BEGIN
1161          UPDATE angajat
1162          SET salariu = 99999
1163
1164      END;
1165
1166      BEGIN
1167          UPDATE angajat
1168          SET salariu = 99999
1169
1170      END;
1171
1172      BEGIN
1173          UPDATE angajat
1174          SET salariu = 99999
1175
1176      END;
1177
1178      BEGIN
1179          UPDATE angajat
1180          SET salariu = 99999
1181
1182      END;
1183
1184      BEGIN
1185          UPDATE angajat
1186          SET salariu = 99999
1187
1188      END;
1189
1190      BEGIN
1191          UPDATE angajat
1192          SET salariu = 99999
1193
1194      END;
1195
1196      BEGIN
1197          UPDATE angajat
1198          SET salariu = 99999
1199
1200      END;
1201
1202      BEGIN
1203          UPDATE angajat
1204          SET salariu = 99999
1205
1206      END;
1207
1208      BEGIN
1209          UPDATE angajat
1210          SET salariu = 99999
1211
1212      END;
1213
1214      BEGIN
1215          UPDATE angajat
1216          SET salariu = 99999
1217
1218      END;
1219
1220      BEGIN
1221          UPDATE angajat
1222          SET salariu = 99999
1223
1224      END;
1225
1226      BEGIN
1227          UPDATE angajat
1228          SET salariu = 99999
1229
1230      END;
1231
1232      BEGIN
1233          UPDATE angajat
1234          SET salariu = 99999
1235
1236      END;
1237
1238      BEGIN
1239          UPDATE angajat
1240          SET salariu = 99999
1241
1242      END;
1243
1244      BEGIN
1245          UPDATE angajat
1246          SET salariu = 99999
1247
1248      END;
1249
1250      BEGIN
1251          UPDATE angajat
1252          SET salariu = 99999
1253
1254      END;
1255
1256      BEGIN
1257          UPDATE angajat
1258          SET salariu = 99999
1259
1260      END;
1261
1262      BEGIN
1263          UPDATE angajat
1264          SET salariu = 99999
1265
1266      END;
1267
1268      BEGIN
1269          UPDATE angajat
1270          SET salariu = 99999
1271
1272      END;
1273
1274      BEGIN
1275          UPDATE angajat
1276          SET salariu = 99999
1277
1278      END;
1279
1280      BEGIN
1281          UPDATE angajat
1282          SET salariu = 99999
1283
1284      END;
1285
1286      BEGIN
1287          UPDATE angajat
1288          SET salariu = 99999
1289
1290      END;
1291
1292      BEGIN
1293          UPDATE angajat
1294          SET salariu = 99999
1295
1296      END;
1297
1298      BEGIN
1299          UPDATE angajat
1300          SET salariu = 99999
1301
1302      END;
1303
1304      BEGIN
1305          UPDATE angajat
1306          SET salariu = 99999
1307
1308      END;
1309
1310      BEGIN
1311          UPDATE angajat
1312          SET salariu = 99999
1313
1314      END;
1315
1316      BEGIN
1317          UPDATE angajat
1318          SET salariu = 99999
1319
1320      END;
1321
1322      BEGIN
1323          UPDATE angajat
1324          SET salariu = 99999
1325
1326      END;
1327
1328      BEGIN
1329          UPDATE angajat
1330          SET salariu = 99999
1331
1332      END;
1333
1334      BEGIN
1335          UPDATE angajat
1336          SET salariu = 99999
1337
1338      END;
1339
1340      BEGIN
1341          UPDATE angajat
1342          SET salariu = 99999
1343
1344      END;
1345
1346      BEGIN
1347          UPDATE angajat
1348          SET salariu = 99999
1349
1350      END;
1351
1352      BEGIN
1353          UPDATE angajat
1354          SET salariu = 99999
1355
1356      END;
1357
1358      BEGIN
1359          UPDATE angajat
1360          SET salariu = 99999
1361
1362      END;
1363
1364      BEGIN
1365          UPDATE angajat
1366          SET salariu = 99999
1367
1368      END;
1369
1370      BEGIN
1371          UPDATE angajat
1372          SET salariu = 99999
1373
1374      END;
1375
1376      BEGIN
1377          UPDATE angajat
1378          SET salariu = 99999
1379
1380      END;
1381
1382      BEGIN
1383          UPDATE angajat
1384          SET salariu = 99999
1385
1386      END;
1387
1388      BEGIN
1389          UPDATE angajat
1390          SET salariu = 99999
1391
1392      END;
1393
1394      BEGIN
1395          UPDATE angajat
1396          SET salariu = 99999
1397
1398      END;
1399
1400      BEGIN
1401          UPDATE angajat
1402          SET salariu = 99999
1403
1404      END;
1405
1406      BEGIN
1407          UPDATE angajat
1408          SET salariu = 99999
1409
1410      END;
1411
1412      BEGIN
1413          UPDATE angajat
1414          SET salariu = 99999
1415
1416      END;
1417
1418      BEGIN
1419          UPDATE angajat
1420          SET salariu = 99999
1421
1422      END;
1423
1424      BEGIN
1425          UPDATE angajat
1426          SET salariu = 99999
1427
1428      END;
1429
1430      BEGIN
1431          UPDATE angajat
1432          SET salariu = 99999
1433
1434      END;
1435
1436      BEGIN
1437          UPDATE angajat
1438          SET salariu = 99999
1439
1440      END;
1441
1442      BEGIN
1443          UPDATE angajat
1444          SET salariu = 99999
1445
1446      END;
1447
1448      BEGIN
1449          UPDATE angajat
1450          SET salariu = 99999
1451
1452      END;
1453
1454      BEGIN
1455          UPDATE angajat
1456          SET salariu = 99999
1457
1458      END;
1459
1460      BEGIN
1461          UPDATE angajat
1462          SET salariu = 99999
1463
1464      END;
1465
1466      BEGIN
1467          UPDATE angajat
1468          SET salariu = 99999
1469
1470      END;
1471
1472      BEGIN
1473          UPDATE angajat
1474          SET salariu = 99999
1475
1476      END;
1477
1478      BEGIN
1479          UPDATE angajat
1480          SET salariu = 99999
1481
1482      END;
1483
1484      BEGIN
1485          UPDATE angajat
1486          SET salariu = 99999
1487
1488      END;
1489
1490      BEGIN
1491          UPDATE angajat
1492          SET salariu = 99999
1493
1494      END;
1495
1496      BEGIN
1497          UPDATE angajat
1498          SET salariu = 99999
1499
1500      END;
1501
1502      BEGIN
1503          UPDATE angajat
1504          SET salariu = 99999
1505
1506      END;
1507
1508      BEGIN
1509          UPDATE angajat
1510          SET salariu = 99999
1511
1512      END;
1513
1514      BEGIN
1515          UPDATE angajat
1516          SET salariu = 99999
1517
1518      END;
1519
1520      BEGIN
1521          UPDATE angajat
1522          SET salariu = 99999
1523
1524      END;
1525
1526      BEGIN
1527          UPDATE angajat
1528          SET salariu = 99999
1529
1530      END;
1531
1532      BEGIN
1533          UPDATE angajat
1534          SET salariu = 99999
1535
1536      END;
1537
1538      BEGIN
1539          UPDATE angajat
1540          SET salariu = 99999
1541
1542      END;
1543
1544      BEGIN
1545          UPDATE angajat
1546          SET salariu = 99999
1547
1548      END;
1549
1550      BEGIN
1551          UPDATE angajat
1552          SET salariu = 99999
1553
1554      END;
1555
1556      BEGIN
1557          UPDATE angajat
1558          SET salariu = 99999
1559
1560      END;
1561
1562      BEGIN
1563          UPDATE angajat
1564          SET salariu = 99999
1565
1566      END;
1567
1568      BEGIN
1569          UPDATE angajat
1570          SET salariu = 99999
1571
1572      END;
1573
1574      BEGIN
1575          UPDATE angajat
1576          SET salariu = 99999
1577
1578      END;
1579
1580      BEGIN
1581          UPDATE angajat
1582          SET salariu = 99999
1583
1584      END;
1585
1586      BEGIN
1587          UPDATE angajat
1588          SET salariu = 99999
1589
1590      END;
1591
1592      BEGIN
1593          UPDATE angajat
1594          SET salariu = 99999
1595
1596      END;
1597
1598      BEGIN
1599          UPDATE angajat
1600          SET salariu = 99999
1601
1602      END;
1603
1604      BEGIN
1605          UPDATE angajat
1606          SET salariu = 99999
1607
1608      END;
1609
1610      BEGIN
1611          UPDATE angajat
1612          SET salariu = 99999
1613
1614      END;
1615
1616      BEGIN
1617          UPDATE angajat
1618          SET salariu = 99999
1619
1620      END;
1621
1622      BEGIN
1623          UPDATE angajat
1624          SET salariu = 99999
1625
1626      END;
1627
1628      BEGIN
1629          UPDATE angajat
1630          SET salariu = 99999
1631
1632      END;
1633
1634      BEGIN
1635          UPDATE angajat
1636          SET salariu = 99999
1637
1638      END;
1639
1640      BEGIN
1641          UPDATE angajat
1642          SET salariu = 99999
1643
1644      END;
1645
1646      BEGIN
1647          UPDATE angajat
1648          SET salariu = 99999
1649
1650      END;
1651
1652      BEGIN
1653          UPDATE angajat
1654          SET salariu = 99999
1655
1656      END;
1657
1658      BEGIN
1659          UPDATE angajat
1660          SET salariu = 99999
1661
1662      END;
1663
1664      BEGIN
1665          UPDATE angajat
1666          SET salariu = 99999
1667
1668      END;
1669
1670      BEGIN
1671          UPDATE angajat
1672          SET salariu = 99999
1673
1674      END;
1675
1676      BEGIN
1677          UPDATE angajat
1678          SET salariu = 99999
1679
1680      END;
1681
1682      BEGIN
1683          UPDATE angajat
1684          SET salariu = 99999
1685
1686      END;
1687
1688      BEGIN
1689          UPDATE angajat
1690          SET salariu = 99999
1691
1692      END;
1693
1694      BEGIN
1695          UPDATE angajat
1696          SET salariu = 99999
1697
1698      END;
1699
1700      BEGIN
1701          UPDATE angajat
1702          SET salariu = 99999
1703
1704      END;
1705
1706      BEGIN
1707          UPDATE angajat
1708          SET salariu = 99999
1709
1710      END;
1711
1712      BEGIN
1713          UPDATE angajat
1714          SET salariu = 99999
1715
1716      END;
1717
1718      BEGIN
1719          UPDATE angajat
1720          SET salariu = 99999
1721
1722      END;
1723
1724      BEGIN
1725          UPDATE angajat
1726          SET salariu = 99999
1727
1728      END;
1729
1730      BEGIN
1731          UPDATE angajat
1732          SET salariu = 99999
1733
1734      END;
1735
1736      BEGIN
1737          UPDATE angajat
1738          SET salariu = 99999
1739
1740      END;
1741
1742      BEGIN
1743          UPDATE angajat
1744          SET salariu = 99999
1745
1746      END;
1747
1748      BEGIN
1749          UPDATE angajat
1750          SET salariu = 99999
1751
1752      END;
1753
1754      BEGIN
1755          UPDATE angajat
1756          SET salariu = 99999
1757
1758      END;
1759
1760      BEGIN
1761          UPDATE angajat
1762          SET salariu = 99999
1763
1764      END;
1765
1766      BEGIN
1767          UPDATE angajat
1768          SET salariu = 99999
1769
1770      END;
1771
1772      BEGIN
1773          UPDATE angajat
1774          SET salariu = 99999
1775
1776      END;
1777
1778      BEGIN
1779          UPDATE angajat
1780          SET salariu = 99999
1781
1782      END;
1783
1784      BEGIN
1785          UPDATE angajat
1786          SET salariu = 99999
1787
1788      END;
1789
1790      BEGIN
1791          UPDATE angajat
1792          SET salariu = 99999
1793
1794      END;
1795
1796      BEGIN
1797          UPDATE angajat
1798          SET salariu = 99999
1799
1800      END;
1801
1802      BEGIN
1803          UPDATE angajat
1804          SET salariu = 99999
1805
1806      END;
1807
1808      BEGIN
1809          UPDATE angajat
1810          SET salariu = 99999
1811
1812      END;
1813
1814      BEGIN
1815          UPDATE angajat
1816          SET salariu = 99999
1817
1818      END;
1819
1820      BEGIN
1821          UPDATE angajat
1822          SET salariu = 99999
1823
1824      END;
1825
1826      BEGIN
1827          UPDATE angajat
1828          SET salariu = 99999
1829
1830      END;
1831
1832      BEGIN
1833         
```

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** Shows "sql/project" and "Version control".
- Database Explorer:** On the left, under "C##PROJECT_SGBD", there are nodes for "tables" (14), "columns" (8), "keys" (1), "foreign keys" (2), and "indexes" (1). The "FACILITATE" node is currently selected.
- SQL Editor:** The main area displays a SQL script named "hoteluri_cu_modificari". The code includes several UPDATE statements, IF blocks, and DBMS_OUTPUT.PUT_LINE statements. It also features a BEGIN block with a cursor declaration and a loop that fetches data from a cursor into variables.
- Status Bar:** At the bottom, it shows "498:58 CRLF UTF-8 4 spaces".
- Bottom Icons:** Includes icons for search, browser, file, and various development tools.

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          COD_ANGA
          COD_HOTEL
          NUME VARCHAR
          PRENUME
          DATA_NAST
          JOB VARCHAR
          SALARIU NUMBER
          COD_MANAGER
        > keys 1
        > foreign keys 2
        > indexes 1
      > APROVIZIONARE
      > CAMERA
      > CLIENT
      > ERORI
      > FACILITATE
      > HOTEL
      > LOCATIE
      > OFERTA
      > PACHET
        > columns 4
          COD_PACHE

```

console_8 [project_sgbd] x console_7 [project_sgbd]

```

381   EXIT WHEN c%NOTFOUND;
382   DBMS_OUTPUT.PUT_LINE( A_ nume_hotel_current);
383   END LOOP;
384   END IF;
385   END;
386
387   DECLARE
388     c SYS_REFCURSOR;
389     cod_hotel_current hotel.cod_hotel%TYPE;
390     nume_hotel_current hotel.nume%TYPE;
391   BEGIN
392     c := hoteluri_pachet_ex14.hoteluri_cu_modificari( TIP_CAUTARE: -1);
393     IF c IS NOT NULL THEN
394       LOOP
395         FETCH c INTO cod_hotel_current, nume_hotel_current;
396         EXIT WHEN c%NOTFOUND;
397         DBMS_OUTPUT.PUT_LINE( A_ nume_hotel_current);
398       END LOOP;
399     END IF;
400   END;
401
402   DECLARE
403     c SYS_REFCURSOR;
404     cod_hotel_current hotel.cod_hotel%TYPE;
405     nume_hotel_current hotel.nume%TYPE;
406   BEGIN
407     c := hoteluri_pachet_ex14.hoteluri_cu_modificari( TIP_CAUTARE: 0);
408     IF c IS NOT NULL THEN
409       LOOP

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

498:58 CRLF UTF-8 4 spaces ⌂ ENG 12:24 AM 1/11/2024

Database Explorer

```

> project 2 of 40
> project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 14
      > ANEXA
      > ANGAJAT
        > columns 8
          COD_ANGA
          COD_HOTEL
          NUME VARCHAR
          PRENUME
          DATA_NAST
          JOB VARCHAR
          SALARIU NUMBER
          COD_MANAGER
        > keys 1
        > foreign keys 2
        > indexes 1
      > APROVIZIONARE
      > CAMERA
      > CLIENT
      > ERORI
      > FACILITATE
      > HOTEL
      > LOCATIE
      > OFERTA
      > PACHET
        > columns 4
          COD_PACHE

```

console_8 [project_sgbd] x console_7 [project_sgbd]

```

409     IF c IS NOT NULL THEN
410       LOOP
411         FETCH c INTO cod_hotel_current, nume_hotel_current;
412         EXIT WHEN c%NOTFOUND;
413         DBMS_OUTPUT.PUT_LINE( A_ nume_hotel_current);
414       END LOOP;
415     END IF;
416   END;
417
418   DBMS_OUTPUT.PUT_LINE( A_ '*****');
419
420   BEGIN
421     BEGIN
422       hoteluri_pachet_ex14.marire_salarii_angajati( COD_HOTEL_DORIT: 0);
423       ROLLBACK;
424     END;
425
426     BEGIN
427       UPDATE angajat
428       SET salariu = 9999
429       WHERE cod_hotel = 9 AND cod_manager IS NOT NULL;
430
431       hoteluri_pachet_ex14.marire_salarii_angajati( COD_HOTEL_DORIT: 0);
432       ROLLBACK;
433     END;
434   END;
435
436   DBMS_OUTPUT.PUT_LINE( A_ '*****');

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

498:58 CRLF UTF-8 4 spaces ⌂ ENG 12:24 AM 1/11/2024

sqlproject Version control

Database Explorer

console_8 [project_sgbdb] x console_7 [project_sgbdb]

File Edit View Insert Tools Window Help

Project 2 of 40 project_sgbdb 2 of 41

C# #PROJECT_SGBD

tables 14 ANEXA ANGAJAT columns 8 COD_ANGA COD_HOTEL NUME VARD PRENUME DATA_NAST JOB VARCH SALARIU NUM COD_MANAGER keys 1 foreign keys 2 indexes 1 APROVIZIONARE CAMERA CLIENT ERORI FACILITATE HOTEL LOCATIE OFERTA PACHET columns 4 COD_PACHE

438 BEGIN hoteluri_pachet_ex14.scadere_preturi_pachete(COD_HOTEL_DORIT, 9); ROLLBACK; END;

439

440

441

442

443 DBMS_OUTPUT.PUT_LINE(A: '*****');

444 BEGIN

445 BEGIN hoteluri_pachet_ex14.bonus_manageri; ROLLBACK; END;

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

BEGIN UPDATE angajat SET salariu = 1000 WHERE cod_manager IS NOT NULL; hoteluri_pachet_ex14.bonus_manageri; ROLLBACK; END;

DBMS_OUTPUT.PUT_LINE(A: '*****');

BEGIN -- marire salarii angajati BEGIN hoteluri_pachet_ex14.verificare_trimestrialia;

ROLLBACK; END;

DBMS_OUTPUT.PUT_LINE(A: '-----');

-- scadere preturi pachete BEGIN INSERT INTO rezervare VALUES (19, 106, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 99999); hoteluri_pachet_ex14.verificare_trimestrialia; ROLLBACK; END;

DBMS_OUTPUT.PUT_LINE(A: '-----');

-- bonus manageri BEGIN INSERT INTO rezervare VALUES (19, 106, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 10720-1800); hoteluri_pachet_ex14.verificare_trimestrialia; ROLLBACK; END;

DBMS_OUTPUT.PUT_LINE(A: '-----');

503:2 CRLF UTF-8 4 spaces ⌂ 12:44 AM 1/11/2024

sqlproject Version control

Database Explorer

console_8 [project_sgbdb] x console_7 [project_sgbdb]

File Edit View Insert Tools Window Help

Project 2 of 40 project_sgbdb 2 of 41

C# #PROJECT_SGBD

tables 14 ANEXA ANGAJAT columns 8 COD_ANGA COD_HOTEL NUME VARD PRENUME DATA_NAST JOB VARCH SALARIU NUM COD_MANAGER keys 1 foreign keys 2 indexes 1 APROVIZIONARE CAMERA CLIENT ERORI FACILITATE HOTEL LOCATIE OFERTA PACHET columns 4 COD_PACHE

401 DBMS_OUTPUT.PUT_LINE(A: '*****');

402

403

404

405

406

407

408

409

410

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

491

492

493

494

495

496

497

498

499

500

402 BEGIN -- marire salarii angajati BEGIN hoteluri_pachet_ex14.verificare_trimestrialia;

ROLLBACK; END;

403 DBMS_OUTPUT.PUT_LINE(A: '-----');

404 -- scadere preturi pachete BEGIN

405 INSERT INTO rezervare VALUES (19, 106, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 99999);

406 hoteluri_pachet_ex14.verificare_trimestrialia;

407 ROLLBACK; END;

408

409 DBMS_OUTPUT.PUT_LINE(A: '-----');

410 -- bonus manageri BEGIN

411 INSERT INTO rezervare VALUES (19, 106, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 10720-1800);

412 hoteluri_pachet_ex14.verificare_trimestrialia;

413 ROLLBACK; END;

414

415 DBMS_OUTPUT.PUT_LINE(A: '-----');

503:2 CRLF UTF-8 4 spaces ⌂ 12:44 AM 1/11/2024

Database Explorer

```

    > project 2 of 40
    > project_sgbd 2 of 41
      > C##PROJECT_SGBD
        > tables 14
          > ANEXA
          > ANGAJAT
            > columns 8
              > COD_ANGAJAT
              > COD_HOTEL
              > NUME VARCHAR
              > PRENUME
              > DATA_NASTERII DATE
              > JOB VARCHAR
              > SALARIU NUMBER
              > COD_MANAGER
            > keys 1
            > foreign keys 2
            > indexes 1
          > APROVIZIONARE
          > CAMERA
          > CLIENT
          > ERORI
          > FACILITATE
          > HOTEL
          > LOCATIE
          > OFERTA
          > PACHET
            > columns 4
              > COD_PACHET

```

```

474 BEGIN
475   INSERT INTO rezervare VALUES (19, 106, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 99999);
476
477   hoteluri_pachet_ex14.verificare_trimestrial;
478   ROLLBACK;
479 END;

DBMS_OUTPUT.PUT_LINE('-----');

-- bonus manageri
BEGIN
  INSERT INTO rezervare VALUES (19, 106, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 10720-1800);

  hoteluri_pachet_ex14.verificare_trimestrial;
  ROLLBACK;
END;

DBMS_OUTPUT.PUT_LINE('-----');

-- exc_perioada
BEGIN
  DELETE FROM rezervare
  WHERE EXTRACT(YEAR FROM data_sosire) = 2021
    AND EXTRACT(MONTH FROM data_sosire) = 1 OR EXTRACT(MONTH FROM data_sosire) = 2 OR EXTRACT(MONTH FROM data_sosire) = 3;

  hoteluri_pachet_ex14.verificare_trimestrial;
  ROLLBACK;
END;
END;

```

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Partly cloudy

503:2 CRLF UTF-8 4 spaces ⌂ 1245 AM 1/11/2024

Database Explorer

```

    > project 2 of 40
    > project_sgbd 2 of 41
      > C##PROJECT_SGBD
        > tables 14

```

Services

	COD_ANGAJAT	COD_HOTEL	NUME	PRENUME	DATA_NASTERII	JOB	SALARIU	COD_MANAGER
19	96	8 Stan	Alexia	1994-12-28	Menajera	3800	71	
20	72	9 Radu	Mihai	1998-05-27	Manager	27000	<null>	
21	97	9 Angelina	George	1989-03-10	Receptioner	5500	72	
22	98	9 Sandu	Alin	1980-05-07	Bucatar	7500	72	
23	99	9 Cristea	Dumitru	1984-06-08	Chelner	4000	72	
24	100	9 Dumitrescu	Raluca	1982-12-26	Chelner	4100	72	
25	101	9 Stanescu	Ana	1993-07-03	Menajera	4000	72	
26	102	9 Enache	Corina	1996-09-23	Menajera	3900	72	
27	65	10 Constantinescu	Diana	1988-05-27	Manager	35000	<null>	
28	66	10 Pop	Adrian	1990-08-18	Manager	30000	<null>	
29	67	10 Ionescu	Radu	1991-07-15	Manager	33000	<null>	
30	73	10 Nistor	David	1988-04-09	Receptioner	6500	65	
31	74	10 Calinescu	Cosmin	1989-05-10	Receptioner	6100	65	
32	75	10 Zaharia	Marius	1987-11-16	Bucatar	8500	67	
33	76	10 Toma	Andrei	1997-10-07	Chelner	4780	67	
34	77	10 Stanciu	Stefan	1991-02-25	Chelner	4900	67	
35	78	10 Munteanu	Mihai	1995-08-09	Chelner	5200	67	
36	79	10 Neagu	Oana	1996-09-23	Menajera	3975	66	
37	80	10 Cojocaru	Cosmina	1995-06-18	Menajera	3950	66	
38	81	10 Ionita	Irina	1996-08-24	Menajera	3900	66	

Database Consoles > project_sgbd > console_8 [project_sgbd]

-5°C Cold weather

510:9 CRLF UTF-8 4 spaces ⌂ 1229 AM 1/11/2024

Database Explorer

Console_8 [project_sgbdb] × Console_7 [project_sgbdb]

Services

Output C#PROJECT_SGBD.ANGAJAT

COD_ANGAJAT	COD_HOTEL	NUME	PRENUME	DATA_NASTERII	JOB	SALARIU	COD_MANAGER
19	96	8 Stan	Alexia	1994-12-28	Menajera	3800	71
20	72	9 Radu	Mihai	1990-05-27	Manager	27000	<null>
21	97	9 Anghelina	George	1989-03-10	Receptioner	5500	72
22	98	9 Sandu	Alin	1988-05-07	Bucatar	7500	72
23	99	9 Cristea	Dumitru	1984-06-08	Chelner	4500	72
24	100	9 Dumitrescu	Raluca	1982-12-26	Chelner	4600	72
25	101	9 Stanescu	Ana	1993-07-03	Menajera	4500	72
26	102	9 Enache	Corina	1996-09-23	Menajera	4400	72
27	65	10 Constantinescu	Diana	1988-05-27	Manager	35000	<null>
28	66	10 Pop	Adrian	1990-08-18	Manager	30000	<null>
29	67	10 Ionescu	Radu	1991-07-15	Manager	33000	<null>
30	73	10 Mistor	David	1988-04-09	Receptioner	7000	65
31	74	10 Calinescu	Cosmin	1989-03-10	Receptioner	6600	65
32	75	10 Zaharia	Marius	1987-11-16	Bucatar	8500	67
33	76	10 Toma	Andrei	1997-10-07	Chelner	5280	67
34	77	10 Stanciu	Stefan	1991-02-25	Chelner	5400	67
35	78	10 Munteanu	Mihai	1995-08-09	Chelner	5700	67
36	79	10 Neagu	Oana	1996-09-23	Menajera	4475	66
37	80	10 Cojocaru	Cosmina	1995-06-18	Menajera	4450	66
38	81	10 Ionita	Irina	1996-08-24	Menajera	4400	66

Database Consoles > project_sgbdb > Console_8 [project_sgbdb]

510:9 CRLF UTF-8 4 spaces ⌂ ENG 12:29 AM 1/11/2024

Database Explorer

Console_8 [project_sgbdb] × Console_7 [project_sgbdb]

Services

Output C#PROJECT_SGBD.PACHET

COD_PACHET	COD_HOTEL	BENEFICIU	PRET
1	149	10 All Inclusive	500
2	150	10 Acces la Spa si Sala de fitness	300
3	151	10 Acces la Piscina	250
4	152	10 Acces la Teren de sport si Sala de jocuri	200
5	153	10 Tur ghidat	150
6	154	10 Activitati nautice	300
7	155	6 All Inclusive	500
8	156	6 Acces la Spa si Sala de fitness	350
9	157	6 Acces la Piscina	230
10	158	6 Acces la muzeele din oras	200
11	159	6 Tur ghidat	170
12	160	7 All Inclusive	400
13	161	7 Acces la muzeele din oras	140
14	162	7 Acces la Piscina	150
15	163	8 All Inclusive	400
16	164	8 Acces la Piscina	180
17	165	9 All Inclusive	450
18	166	9 Drumetii pe munte	350
19	167	9 Acces la Spa si Sala de fitness	275
20	168	9 Acces la Sala de jocuri	120

Database Consoles > project_sgbdb > Console_8 [project_sgbdb]

513:13 CRLF UTF-8 4 spaces ⌂ ENG 12:31 AM 1/11/2024

Database Explorer

```
SELECT * FROM COD_PACHET
```

	COD_PACHET	COD_HOTEL	BENEFICIU	PRET
1	149	10	All Inclusive	500
2	150	10	Acces la Spa si Sala de fitness	300
3	151	10	Acces la Piscina	250
4	152	10	Acces la Teren de sport si Sala de jocuri	200
5	153	10	Tur ghidat	150
6	154	10	Activitati nautice	300
7	155	6	All Inclusive	470
8	156	6	Acces la Spa si Sala de fitness	320
9	157	6	Acces la Piscina	280
10	158	6	Acces la muzeele din oras	170
11	159	6	Tur ghidat	140
12	160	7	All Inclusive	370
13	161	7	Acces la muzeele din oras	110
14	162	7	Acces la Piscina	120
15	163	8	All Inclusive	370
16	164	8	Acces la Piscina	150
17	165	9	All Inclusive	420
18	166	9	Drumetii pe munte	320
19	167	9	Acces la Spa si Sala de fitness	245
20	168	9	Acces la Sala de jocuri	90

Services

Output C#\#PROJECT_SGBD.PACHET

```
SELECT * FROM COD_PACHET
```

	COD_PACHET	COD_HOTEL	BENEFICIU	PRET
1	149	10	All Inclusive	500
2	150	10	Acces la Spa si Sala de fitness	300
3	151	10	Acces la Piscina	250
4	152	10	Acces la Teren de sport si Sala de jocuri	200
5	153	10	Tur ghidat	150
6	154	10	Activitati nautice	300
7	155	6	All Inclusive	470
8	156	6	Acces la Spa si Sala de fitness	320
9	157	6	Acces la Piscina	280
10	158	6	Acces la muzeele din oras	170
11	159	6	Tur ghidat	140
12	160	7	All Inclusive	370
13	161	7	Acces la muzeele din oras	110
14	162	7	Acces la Piscina	120
15	163	8	All Inclusive	370
16	164	8	Acces la Piscina	150
17	165	9	All Inclusive	420
18	166	9	Drumetii pe munte	320
19	167	9	Acces la Spa si Sala de fitness	245
20	168	9	Acces la Sala de jocuri	90

Database Consoles > project_sgbdb > console_8 [project_sgbdb]

512:9 CRLF UTF-8 4 spaces ENG 12:32 AM 1/11/2024

Database Explorer

```
DELETE FROM ANGAJAT;
```

	COD_ANGAJAT	COD_HOTEL	NUME	PRENUME	DATA_NASTERII	JOB	SALARIU	COD_MANAGER
1	65	10	Constantinescu	Diana	1988-05-27	Manager	35000	<null>
2	66	10	Pop	Adrian	1990-08-18	Manager	30000	<null>
3	67	10	Ionescu	Radu	1991-07-15	Manager	33000	<null>
4	68	6	Popescu	Alexandru	1998-05-27	Manager	32000	<null>
5	69	6	Radulescu	Luca	1990-05-27	Manager	31000	<null>
6	70	7	Marinescu	Andrei	1998-05-27	Manager	18000	<null>
7	71	8	Valeriu	George	1990-05-27	Manager	22000	<null>
8	72	9	Radu	Mihai	1990-05-27	Manager	27000	<null>
9	73	10	Nistor	David	1988-04-09	Receptioner	6500	65
10	74	10	Calinescu	Cosmin	1989-03-10	Receptioner	6100	65
11	75	10	Zaharia	Marius	1987-11-16	Bucatar	8500	67
12	76	10	Toma	Andrei	1997-10-07	Chelner	4700	67
13	77	10	Stanciu	Stefan	1991-02-25	Chelner	4900	67
14	78	10	Munteanu	Mihai	1995-08-09	Chelner	5200	67

Services

Output C#\#PROJECT_SGBD.ANGAJAT

```
DELETE FROM ANGAJAT;
```

	COD_ANGAJAT	COD_HOTEL	NUME	PRENUME	DATA_NASTERII	JOB	SALARIU	COD_MANAGER
1	65	10	Constantinescu	Diana	1988-05-27	Manager	35000	<null>
2	66	10	Pop	Adrian	1990-08-18	Manager	30000	<null>
3	67	10	Ionescu	Radu	1991-07-15	Manager	33000	<null>
4	68	6	Popescu	Alexandru	1998-05-27	Manager	32000	<null>
5	69	6	Radulescu	Luca	1990-05-27	Manager	31000	<null>
6	70	7	Marinescu	Andrei	1998-05-27	Manager	18000	<null>
7	71	8	Valeriu	George	1990-05-27	Manager	22000	<null>
8	72	9	Radu	Mihai	1990-05-27	Manager	27000	<null>
9	73	10	Nistor	David	1988-04-09	Receptioner	6500	65
10	74	10	Calinescu	Cosmin	1989-03-10	Receptioner	6100	65
11	75	10	Zaharia	Marius	1987-11-16	Bucatar	8500	67
12	76	10	Toma	Andrei	1997-10-07	Chelner	4700	67
13	77	10	Stanciu	Stefan	1991-02-25	Chelner	4900	67
14	78	10	Munteanu	Mihai	1995-08-09	Chelner	5200	67

Database Consoles > project_sgbdb > console_8 [project_sgbdb]

513:14 CRLF UTF-8 4 spaces ENG 12:43 AM 1/11/2024

sqiproject Version control

Database Explorer Database Explorer Tx: Manual C# PROJECT_SGBD console_8

project [2 of 40] project_sgbd [2 of 41] C# PROJECT_SGBD

tables 14 ANEXA ANGAJAT columns 8 COD_ANGA COD_HOTEL NUME VARD

--ROLLBACK;

511 EXEC;

512 ✓ SELECT * FROM angajat;

513 ROLLBACK;

514

515

516

Services Tx Output C# PROJECT_SGBD.ANGAJAT

Database Database

project_sgbd default 10 ms

console_4

console_5

console_2

console_1

console_8 86 ms

console_7

console_6

console_3

viad

console_1

38 rows

	COD_ANGAJAT	COD_HOTEL	NUME	PRENUME	DATA_NASTERII	JOB	SALARIU	COD_MANAGER
1	65	10	Constantinescu	Diana	1988-05-27	Manager	35610	<null>
2	66	10	Pop	Adrian	1990-08-18	Manager	30000	<null>
3	67	10	Ionescu	Radu	1991-07-15	Manager	33478	<null>
4	68	6	Popescu	Alexandru	1990-05-27	Manager	32575	<null>
5	69	6	Radulescu	Luca	1990-05-27	Manager	31000	<null>
6	70	7	Marinescu	Andrei	1990-05-27	Manager	18340	<null>
7	71	8	Valeriu	George	1990-05-27	Manager	22380	<null>
8	72	9	Radu	Mihai	1990-05-27	Manager	27390	<null>
9	73	10	Nistor	David	1988-04-09	Receptioner	6500	65
10	74	10	Calinescu	Cosmin	1989-03-10	Receptioner	6100	65
11	75	10	Zaharia	Marius	1987-11-16	Bucatar	8500	67
12	76	10	Toma	Andrei	1997-10-07	Chelner	4780	67
13	77	10	Stanciu	Stefan	1991-02-25	Chelner	4900	67
14	78	10	Munteanu	Mihai	1995-08-09	Chelner	5200	67
...	...	10	Mocanu	Oana	1994-09-23	Manager	7025	67

512:7 CRLF UTF-8 4 spaces 12:43 AM 1/11/2024

-5°C Partly cloudy