

Proiect - Sisteme de Gestiune a Bazelor de Date

Gestiunea unui lanț hotelier

Ciuperceanu Vlad-Mihai

Grupa 251

1. Prezentați pe scurt baza de date (utilitatea ei).....	3
2. Realizați diagrama entitate-relație (ERD).....	4
3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.....	5
4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).....	6
5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).....	9
6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate.....	18
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor.....	18
8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii.....	18
9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS.....	19
10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.....	19
11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.....	19
12. Definiți un trigger de tip LDD. Declanșați trigger-ul.....	19
13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.....	19
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).....	19

1. Prezentați pe scurt baza de date (utilitatea ei).

Modelul de date va gestiona informații legate de modalitatea de funcționare a unui lanț hotelier, reținând informații atât despre structura internă a unui hotel, cât și despre parteneriatele acestuia și rezervările făcute la hotelul respectiv.

Fiecare hotel se află la o locație și are mai multe anexe proprii, ce propun servicii destinate clientilor. Pentru întreținerea și buna funcționare a acestora, hotelurile pot fi aprovizionate de mai mulți parteneri ce pun la dispoziție anumite tipuri de produse contra cost.

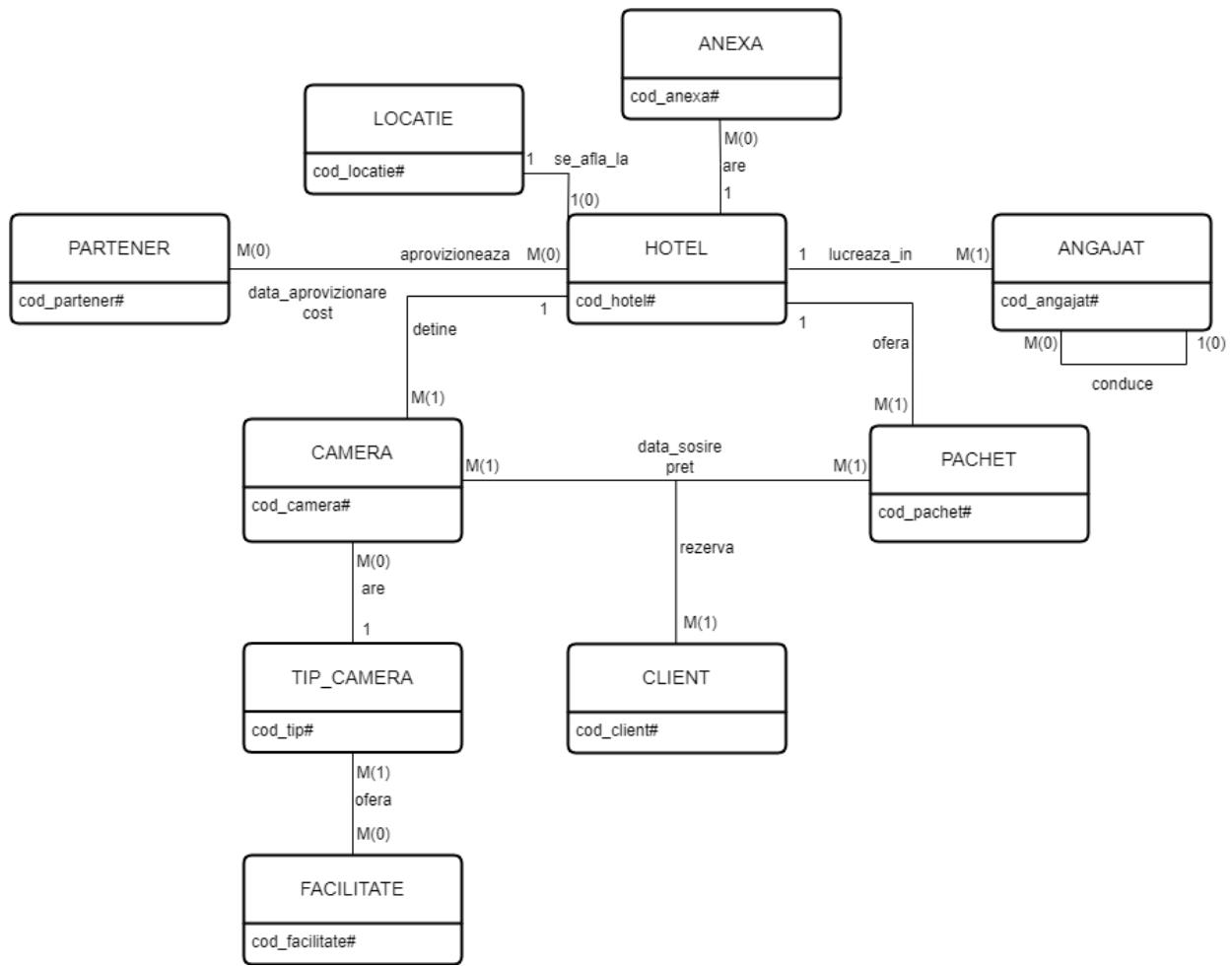
De asemenea, fiecare hotel oferă anumite pachete de cumpărare, ce prezintă beneficii pentru clienti, legate de serviciile angajaților și de folosirea anexelor. Acestea joacă un rol important și în crearea unei rezervări, întrucât clientii trebuie să aleagă un pachet împreună cu o cameră pentru a-și finaliza cazarea.

În cadrul unui hotel lucrează mai mulți angajați, având diverse joburi. Există și o ierarhie printre aceștia, fiind unii angajați care au alți angajați în subordine.

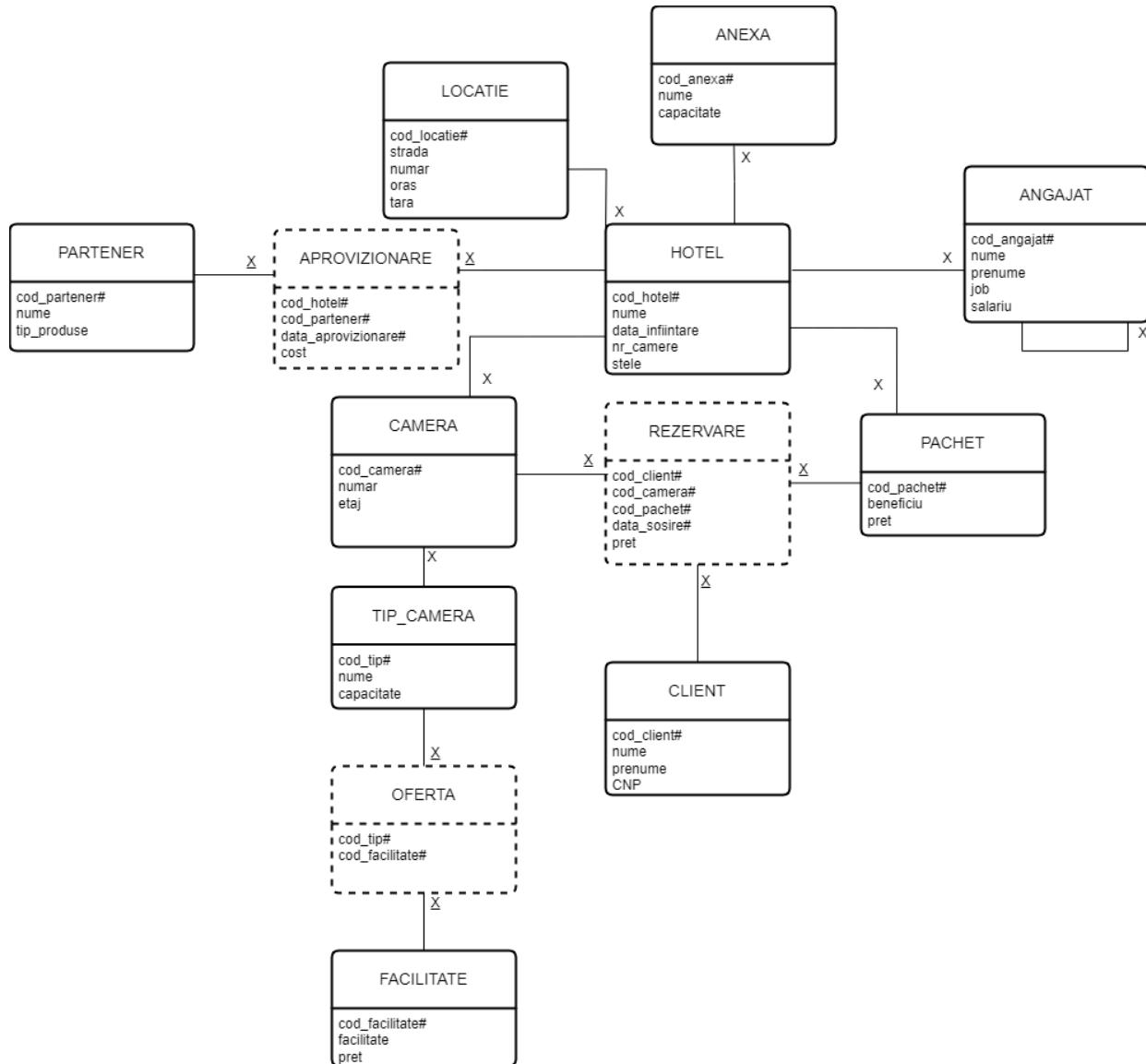
Toate hotelurile au mai multe camere, ce pot fi de mai multe tipuri, în funcție de nevoile și dorințele clientilor. În același timp, fiecare tip de cameră oferă anumite facilități, după care se pot ghida clientii în decizia pe care o vor lua.

Rezervările sunt făcute de către clienti, alegându-și camerele și beneficiile dorite în funcție de pachetele puse la dispoziție de către hotelul la care se dorește cazarea. Este nevoie, pentru finalizarea unei rezervări, să existe o cameră și un pachet pentru care a optat clientul, precum și precizarea datei de sosire.

2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizati diagrama conceptuală a modelului propus, integrând toate atributele necesare.



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```
CREATE TABLE LOCATIE (
    cod_locatie NUMBER(2) PRIMARY KEY,
    strada VARCHAR2(25) NOT NULL,
    numar NUMBER(3) NOT NULL,
   oras VARCHAR2(10) NOT NULL,
    tara VARCHAR2(10) NOT NULL
);

CREATE TABLE HOTEL (
    cod_hotel NUMBER(2) PRIMARY KEY,
    cod_locatie NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    data_infiintare DATE DEFAULT SYSDATE,
    nr_camere NUMBER(2),
    stele NUMBER(1),
    CONSTRAINT fk_hotel FOREIGN KEY (cod_locatie) REFERENCES
LOCATIE(cod_locatie)
);

CREATE TABLE ANEXA (
    cod_anexa NUMBER(2) PRIMARY KEY,
    cod_hotel NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    capacitate NUMBER(2),
    CONSTRAINT fk_anexa FOREIGN KEY (cod_hotel) REFERENCES HOTEL(cod_hotel)
);

CREATE TABLE PACHET (
    cod_pachet NUMBER(3) PRIMARY KEY,
    cod_hotel NUMBER(2),
    beneficiu VARCHAR2(50) NOT NULL,
    pret NUMBER(5) DEFAULT 100,
    CONSTRAINT fk_pachet FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel)
);

CREATE TABLE PARTENER (
    cod_partener NUMBER(2) PRIMARY KEY,
```

```

        nume VARCHAR2(25) NOT NULL,
        tip_produse VARCHAR2(25)
);

CREATE TABLE CLIENT (
    cod_client NUMBER(5) PRIMARY KEY,
    nume VARCHAR2(25) NOT NULL,
    prenume VARCHAR2(25),
    CNP CHAR(13) NOT NULL
);

CREATE TABLE FACILITATE (
    cod_facilitate NUMBER(2) PRIMARY KEY,
    facilitate VARCHAR2(25) NOT NULL,
    pret NUMBER(3)
);

CREATE TABLE TIP_CAMERA (
    cod_tip NUMBER(2) PRIMARY KEY,
    nume VARCHAR2(25) NOT NULL,
    capacitate NUMBER(1) NOT NULL,
    suprafata NUMBER(3) NOT NULL,
    pret NUMBER(4) NOT NULL
);

CREATE TABLE ANGAJAT (
    cod_angajat NUMBER(4) PRIMARY KEY,
    cod_hotel NUMBER(2),
    nume VARCHAR2(25) NOT NULL,
    prenume VARCHAR2(25),
    data_nasterii DATE NOT NULL,
    job VARCHAR2(25) NOT NULL,
    salariu NUMBER(5) NOT NULL,
    cod_manager NUMBER(2),
    CONSTRAINT fk_angajat_hotel FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel),
    CONSTRAINT fk_angajat_manager FOREIGN KEY (cod_manager) REFERENCES
ANGAJAT(cod_angajat)
);

CREATE TABLE CAMERA (
    cod_camera NUMBER(4) PRIMARY KEY,
    cod_hotel NUMBER(2),

```

```

cod_tip NUMBER(2),
numar NUMBER(2) NOT NULL,
etaj NUMBER(1) NOT NULL,
CONSTRAINT fk_camera_hotel FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel),
CONSTRAINT fk_camera_tip FOREIGN KEY (cod_tip) REFERENCES
TIP_CAMERA(cod_tip)
);

CREATE TABLE APROVIZIONARE (
cod_partener NUMBER(2),
cod_hotel NUMBER(2),
data_aprovizionare DATE DEFAULT SYSDATE,
cost NUMBER(5),
CONSTRAINT pk_aprovizionare PRIMARY KEY (cod_partener, cod_hotel,
data_aprovizionare),
CONSTRAINT fk_aprovizionare_partener FOREIGN KEY (cod_partener)
REFERENCES PARTENER(cod_partener),
CONSTRAINT fk_aprovizionare_hotel FOREIGN KEY (cod_hotel) REFERENCES
HOTEL(cod_hotel)
);

CREATE TABLE OFERTA (
cod_tip NUMBER(2),
cod_facilitate NUMBER(2),
CONSTRAINT pk_oferta PRIMARY KEY (cod_tip, cod_facilitate),
CONSTRAINT fk_oferta_tip FOREIGN KEY (cod_tip) REFERENCES
TIP_CAMERA(cod_tip),
CONSTRAINT fk_oferta_facilitate FOREIGN KEY (cod_facilitate) REFERENCES
FACILITATE(cod_facilitate)
);

CREATE TABLE REZERVARE (
cod_client NUMBER(5),
cod_camera NUMBER(4),
cod_pachet NUMBER(3),
data_sosire DATE DEFAULT SYSDATE,
pret NUMBER(5) NOT NULL,
CONSTRAINT pk_rezervare PRIMARY KEY (cod_client, cod_camera,
cod_pachet, data_sosire),
CONSTRAINT fk_rezervare_client FOREIGN KEY (cod_client) REFERENCES
CLIENT(cod_client),
CONSTRAINT fk_rezervare_camera FOREIGN KEY (cod_camera) REFERENCES

```

```

CAMERA(cod_camera),
    CONSTRAINT fk_rezervare_pachet FOREIGN KEY (cod_pachet) REFERENCES
PACHET(cod_pachet)
);

```

The screenshot shows a SQL development environment with the following code:

```

92    CONSTRAINT fk_aprovizionare_partener FOREIGN KEY (cod_partener) REFERENCES PARTENER(cod_partener),
93    CONSTRAINT fk_aprovizionare_hotel FOREIGN KEY (cod_hotel) REFERENCES HOTEL(cod_hotel)
94 );
95
96 ✓ CREATE TABLE OFERTA (
97     cod_tip NUMBER(2),
98     cod_facilitate NUMBER(2),
99     CONSTRAINT pk_oferta PRIMARY KEY (cod_tip, cod_facilitate),
100    CONSTRAINT fk_oferta_tip FOREIGN KEY (cod_tip) REFERENCES TIP_CAMERA(cod_tip),
101    CONSTRAINT fk_oferta_facilitate FOREIGN KEY (cod_facilitate) REFERENCES FACILITATE(cod_facilitate)
102 );
103
104 ✓ CREATE TABLE REZERVARE (
105     cod_client NUMBER(5),
106     cod_camera NUMBER(4),
107     cod_pachet NUMBER(3),
108     data_sosire DATE DEFAULT SYSDATE,
109     pret NUMBER(5) NOT NULL,
110     CONSTRAINT pk_rezervare PRIMARY KEY (cod_client, cod_camera, cod_pachet, data_sosire),
111     CONSTRAINT fk_rezervare_client FOREIGN KEY (cod_client) REFERENCES CLIENT(cod_client),
112     CONSTRAINT fk_rezervare_camera FOREIGN KEY (cod_camera) REFERENCES CAMERA(cod_camera),
113     CONSTRAINT fk_rezervare_pachet FOREIGN KEY (cod_pachet) REFERENCES PACHET(cod_pachet)
114 );

```

The code creates two tables, **OFERTA** and **REZERVARE**. The **OFERTA** table has two columns: **cod_tip** and **cod_facilitate**. It has three constraints: a primary key on both columns, a foreign key constraint **fk_oferta_tip** referencing **TIP_CAMERA**, and a foreign key constraint **fk_oferta_facilitate** referencing **FACILITATE**. The **REZERVARE** table has four columns: **cod_client**, **cod_camera**, **cod_pachet**, and **data_sosire**. It has four constraints: a primary key on all four columns, a foreign key constraint **fk_rezervare_client** referencing **CLIENT**, a foreign key constraint **fk_rezervare_camera** referencing **CAMERA**, and a foreign key constraint **fk_rezervare_pachet** referencing **PACHET**.

- Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```

CREATE SEQUENCE chei
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000;

INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Victoriei', 150,
'Bucuresti', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Calea Unirii', 10, 'Craiova',

```

```

'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Dunarii', 7, 'Dubova',
'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Strada Poiana Soarelui', 15,
'Brasov', 'Romania');
INSERT INTO LOCATIE VALUES (chei.NEXTVAL, 'Bulevardul Ferdinand', 10,
'Constanta', 'Romania');

INSERT INTO HOTEL VALUES (chei.NEXTVAL, 1, 'Novotel', TO_DATE('23-10-2019',
'DD-MM-YYYY'), 70, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 2, 'Ramada Plaza',
TO_DATE('21-08-2017', 'DD-MM-YYYY'), 40, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 3, 'Riviera', TO_DATE('02-05-2016',
'DD-MM-YYYY'), 30, 4);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 4, 'Alpin', TO_DATE('27-09-2012',
'DD-MM-YYYY'), 40, 5);
INSERT INTO HOTEL VALUES (chei.NEXTVAL, 5, 'Vega', TO_DATE('15-03-2019',
'DD-MM-YYYY'), 60, 5);

INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Cif', 'Curatenie');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Lidl', 'Alimente');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Somnia', 'Textile');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Mobexpert', 'Mobilier');
INSERT INTO PARTENER VALUES (chei.NEXTVAL, 'Techfit', 'Echipament
sportiv');

INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Popa', 'Florin',
'5014103009878');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Stoica', 'Roxana',
'2980408230012');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Enescu', 'Bogdan',
'1991016190779');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Diaconu', 'Laura',
'6010101013335');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Vasilescu', 'Andrei',
'1970305170569');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Dinu', 'Ana', '2880614110278');
INSERT INTO CLIENT VALUES (chei.NEXTVAL, 'Marin', 'Cristian',
'1960103150375');

```

```
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Single', 1, 20, 200);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Dubla', 2, 28, 350);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Apartament', 4, 60, 450);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Suite', 3, 50, 400);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Penthouse', 7, 120, 800);
INSERT INTO TIP_CAMERA VALUES (chei.NEXTVAL, 'Royal', 5, 80, 600);

INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Terasa', 70);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Vedere panoramica', 65);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Seif', 60);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Cada cu hidromasaj', 75);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Minibar', 30);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Pat suplimentar', 45);
INSERT INTO FACILITATE VALUES (chei.NEXTVAL, 'Aer conditionat', 40);

INSERT INTO OFERTA VALUES (27, 29);
INSERT INTO OFERTA VALUES (27, 30);
INSERT INTO OFERTA VALUES (27, 31);
INSERT INTO OFERTA VALUES (27, 32);
INSERT INTO OFERTA VALUES (27, 33);
INSERT INTO OFERTA VALUES (27, 34);
INSERT INTO OFERTA VALUES (27, 35);
INSERT INTO OFERTA VALUES (28, 29);
INSERT INTO OFERTA VALUES (28, 30);
INSERT INTO OFERTA VALUES (28, 32);
INSERT INTO OFERTA VALUES (28, 33);
INSERT INTO OFERTA VALUES (28, 34);
INSERT INTO OFERTA VALUES (28, 35);
INSERT INTO OFERTA VALUES (25, 29);
INSERT INTO OFERTA VALUES (25, 30);
INSERT INTO OFERTA VALUES (25, 33);
INSERT INTO OFERTA VALUES (25, 34);
INSERT INTO OFERTA VALUES (25, 35);
INSERT INTO OFERTA VALUES (26, 30);
INSERT INTO OFERTA VALUES (26, 33);
INSERT INTO OFERTA VALUES (26, 34);
INSERT INTO OFERTA VALUES (26, 35);
INSERT INTO OFERTA VALUES (24, 30);
INSERT INTO OFERTA VALUES (24, 33);
INSERT INTO OFERTA VALUES (24, 35);
```

```

INSERT INTO OFERTA VALUES (23, 35);

INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Restaurant', 95);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de fitness', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Spa', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Parcare', 40);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Gradina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Teren de sport', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de conferinte', 80);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 10, 'Sala de jocuri', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Restaurant', 95);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Sala de fitness', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Spa', 25);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Parcare', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 6, 'Sala de conferinte', 90);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Restaurant', 70);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Sala de fitness', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Parcare', 35);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 7, 'Sala de conferinte', 90);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Restaurant', 65);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Piscina', NULL);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Sala de fitness', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 8, 'Parcare', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Restaurant', 80);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Sala de fitness', 25);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Spa', 20);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Parcare', 30);
INSERT INTO ANEXA VALUES (chei.NEXTVAL, 9, 'Sala de jocuri', 35);

INSERT INTO APROVIZIONARE VALUES (11, 10, TO_DATE('23-04-2023',
'DD-MM-YYYY'), 1000);
INSERT INTO APROVIZIONARE VALUES (12, 10, TO_DATE('28-05-2023',
'DD-MM-YYYY'), 1500);
INSERT INTO APROVIZIONARE VALUES (13, 10, TO_DATE('05-05-2023',
'DD-MM-YYYY'), 2500);
INSERT INTO APROVIZIONARE VALUES (14, 10, TO_DATE('01-03-2023',
'DD-MM-YYYY'), 4000);
INSERT INTO APROVIZIONARE VALUES (15, 10, TO_DATE('30-01-2023',

```

```

' DD-MM-YYYY' ), 1500);
INSERT INTO APROVIZIONARE VALUES (11, 6, TO_DATE('21-04-2023',
' DD-MM-YYYY' ), 2000);
INSERT INTO APROVIZIONARE VALUES (12, 6, TO_DATE('24-05-2023',
' DD-MM-YYYY' ), 2500);
INSERT INTO APROVIZIONARE VALUES (13, 6, TO_DATE('09-05-2023',
' DD-MM-YYYY' ), 3000);
INSERT INTO APROVIZIONARE VALUES (14, 6, TO_DATE('01-03-2022',
' DD-MM-YYYY' ), 5000);
INSERT INTO APROVIZIONARE VALUES (15, 6, TO_DATE('25-02-2023',
' DD-MM-YYYY' ), 2000);
INSERT INTO APROVIZIONARE VALUES (11, 7, TO_DATE('19-04-2023',
' DD-MM-YYYY' ), 900);
INSERT INTO APROVIZIONARE VALUES (12, 7, TO_DATE('28-05-2023',
' DD-MM-YYYY' ), 1100);
INSERT INTO APROVIZIONARE VALUES (13, 7, TO_DATE('06-03-2023',
' DD-MM-YYYY' ), 1800);
INSERT INTO APROVIZIONARE VALUES (15, 7, TO_DATE('12-01-2023',
' DD-MM-YYYY' ), 800);
INSERT INTO APROVIZIONARE VALUES (11, 8, TO_DATE('26-04-2023',
' DD-MM-YYYY' ), 1200);
INSERT INTO APROVIZIONARE VALUES (12, 8, TO_DATE('22-05-2023',
' DD-MM-YYYY' ), 1700);
INSERT INTO APROVIZIONARE VALUES (13, 8, TO_DATE('05-02-2023',
' DD-MM-YYYY' ), 1400);
INSERT INTO APROVIZIONARE VALUES (15, 8, TO_DATE('29-01-2023',
' DD-MM-YYYY' ), 500);
INSERT INTO APROVIZIONARE VALUES (11, 9, TO_DATE('24-04-2023',
' DD-MM-YYYY' ), 1300);
INSERT INTO APROVIZIONARE VALUES (12, 9, TO_DATE('27-05-2023',
' DD-MM-YYYY' ), 1600);
INSERT INTO APROVIZIONARE VALUES (13, 9, TO_DATE('02-05-2023',
' DD-MM-YYYY' ), 2300);
INSERT INTO APROVIZIONARE VALUES (14, 9, TO_DATE('08-03-2022',
' DD-MM-YYYY' ), 1900);
INSERT INTO APROVIZIONARE VALUES (15, 9, TO_DATE('31-01-2023',
' DD-MM-YYYY' ), 1200);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Constantinescu', 'Diana',
TO_DATE('27-05-1988', 'DD-MM-YYYY'), 'Manager', 35000, NULL);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Pop', 'Adrian',
TO_DATE('18-08-1990', 'DD-MM-YYYY'), 'Manager', 30000, NULL);

```

```

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Ionescu', 'Radu',
TO_DATE('15-07-1991', 'DD-MM-YYYY'), 'Manager', 33000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Popescu', 'Alexandru',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 32000, NULL);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Radulescu', 'Luca',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 31000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Marinescu', 'Andrei',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 18000, NULL);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Valeriu', 'George',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 22000, NULL);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Radu', 'Mihai',
TO_DATE('27-05-1990', 'DD-MM-YYYY'), 'Manager', 27000, NULL);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Nistor', 'David',
TO_DATE('09/04/1988', 'DD-MM-YYYY'), 'Receptioner', 6500, 65);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Calinescu', 'Cosmin',
TO_DATE('10/03/1989', 'DD-MM-YYYY'), 'Receptioner', 6100, 65);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Zaharia', 'Marius',
TO_DATE('16/11/1987', 'DD-MM-YYYY'), 'Bucatar', 8500, 67);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Toma', 'Andrei',
TO_DATE('07/10/1997', 'DD-MM-YYYY'), 'Chelner', 4780, 67);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Stanciu', 'Stefan',
TO_DATE('25/02/1991', 'DD-MM-YYYY'), 'Chelner', 4900, 67);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Munteanu', 'Mihai',
TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 5200, 67);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Neagu', 'Oana',
TO_DATE('23/09/1996', 'DD-MM-YYYY'), 'Menajera', 3975, 66);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Cojocaru', 'Cosmina',
TO_DATE('18/06/1995', 'DD-MM-YYYY'), 'Menajera', 3950, 66);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 10, 'Ionita', 'Irina',
TO_DATE('24/08/1996', 'DD-MM-YYYY'), 'Menajera', 3900, 66);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Avram', 'Cristian',
TO_DATE('30/07/1985', 'DD-MM-YYYY'), 'Receptioner', 5900, 68);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Puiu', 'Mihnea',
TO_DATE('21/01/1982', 'DD-MM-YYYY'), 'Receptioner', 5750, 68);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Albu', 'Victor',
TO_DATE('01/08/1983', 'DD-MM-YYYY'), 'Bucatar', 8000, 68);

```

```

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Florea', 'Marian',
TO_DATE('28/12/1994', 'DD-MM-YYYY'), 'Chelner', 4300, 69);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Tudor', 'David',
TO_DATE('09/08/1995', 'DD-MM-YYYY'), 'Chelner', 4350, 69);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Petrescu', 'Bianca',
TO_DATE('03/07/1993', 'DD-MM-YYYY'), 'Menajera', 3600, 69);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 6, 'Iacob', 'Giorgiana',
TO_DATE('29/12/1991', 'DD-MM-YYYY'), 'Menajera', 3800, 69);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Gavrila', 'Mihai',
TO_DATE('20/01/1987', 'DD-MM-YYYY'), 'Receptioner', 5200, 70);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Ghinea', 'Mihnea',
TO_DATE('22/02/1997', 'DD-MM-YYYY'), 'Bucatar', 6500, 70);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Vasile', 'Tudor',
TO_DATE('25/07/1995', 'DD-MM-YYYY'), 'Chelner', 3800, 70);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 7, 'Mihai', 'Alexandra',
TO_DATE('12/04/1983', 'DD-MM-YYYY'), 'Menajera', 3400, 70);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Iovan', 'Gabriel',
TO_DATE('17/08/1989', 'DD-MM-YYYY'), 'Receptioner', 5350, 71);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Gheorghe', 'Roxana',
TO_DATE('26/09/1981', 'DD-MM-YYYY'), 'Bucatar', 7000, 71);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Hristea', 'Sebastian',
TO_DATE('12/05/1990', 'DD-MM-YYYY'), 'Chelner', 3900, 71);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 8, 'Stan', 'Alexia',
TO_DATE('28/12/1994', 'DD-MM-YYYY'), 'Menajera', 3800, 71);

INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Anghelina', 'George',
TO_DATE('10/03/1989', 'DD-MM-YYYY'), 'Receptioner', 5500, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Sandu', 'Alin',
TO_DATE('07/05/1980', 'DD-MM-YYYY'), 'Bucatar', 7500, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Cristea', 'Dumitru',
TO_DATE('08/06/1984', 'DD-MM-YYYY'), 'Chelner', 4000, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Dumitrescu', 'Raluca',
TO_DATE('26/12/1982', 'DD-MM-YYYY'), 'Chelner', 4100, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Stanescu', 'Ana',
TO_DATE('03/07/1993', 'DD-MM-YYYY'), 'Menajera', 4000, 72);
INSERT INTO ANGAJAT VALUES (chei.NEXTVAL, 9, 'Enache', 'Corina',
TO_DATE('23/09/1996', 'DD-MM-YYYY'), 'Menajera', 3900, 72);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 27, 80, 8);

```

```
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 27, 72, 7);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 55, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 63, 6);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 28, 61, 6);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 25, 20, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 25, 28, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 26, 29, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 26, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 24, 18, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 24, 14, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 23, 13, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 10, 23, 12, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 27, 66, 6);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 28, 54, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 28, 56, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 25, 44, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 25, 38, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 26, 41, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 26, 37, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 27, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 24, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 23, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 6, 23, 21, 2);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 27, 56, 5);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 28, 44, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 25, 32, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 26, 48, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 26, 37, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 24, 24, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 25, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 9, 23, 17, 1);

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 25, 41, 4);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 31, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 26, 35, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 21, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 24, 17, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 22, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 7, 23, 10, 1);
```

```

INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 25, 34, 3);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 26, 29, 2);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 13, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 24, 12, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 11, 1);
INSERT INTO CAMERA VALUES (chei.NEXTVAL, 8, 23, 10, 1);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'All Inclusive', 500);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Spa si Sala de fitness', 300);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Piscina', 250);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Acces la Teren de sport si Sala de jocuri', 200);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Tur ghidat', 150);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 10, 'Activitati nautice', 300);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'All Inclusive', 500);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Spa si Sala de fitness', 350);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la Piscina', 230);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Acces la muzeele din oras', 200);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 6, 'Tur ghidat', 170);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'All Inclusive', 400);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la muzeele din oras', 140);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 7, 'Acces la Piscina', 150);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'All Inclusive', 400);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 8, 'Acces la Piscina', 180);

INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'All Inclusive', 450);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Drumetii pe munte', 350);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Spa si Sala de fitness', 275);
INSERT INTO PACHET VALUES (chei.NEXTVAL, 9, 'Acces la Sala de jocuri', 120);

INSERT INTO REZERVARE VALUES (16, 136, 161, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 1100);

```

```
INSERT INTO REZERVARE VALUES (16, 109, 153, TO_DATE('21-04-2020',
'DD-MM-YYYY'), 1800);

INSERT INTO REZERVARE VALUES (20, 104, 149, TO_DATE('23-11-2019',
'DD-MM-YYYY'), 2300);
INSERT INTO REZERVARE VALUES (20, 109, 150, TO_DATE('05-06-2020',
'DD-MM-YYYY'), 4400);
INSERT INTO REZERVARE VALUES (20, 128, 166, TO_DATE('09-08-2020',
'DD-MM-YYYY'), 5005);
INSERT INTO REZERVARE VALUES (20, 107, 149, TO_DATE('19-03-2021',
'DD-MM-YYYY'), 6700);
INSERT INTO REZERVARE VALUES (20, 115, 154, TO_DATE('27-10-2021',
'DD-MM-YYYY'), 3200);

INSERT INTO REZERVARE VALUES (17, 143, 164, TO_DATE('23-12-2020',
'DD-MM-YYYY'), 4500);
INSERT INTO REZERVARE VALUES (17, 138, 163, TO_DATE('08-02-2021',
'DD-MM-YYYY'), 3020);
INSERT INTO REZERVARE VALUES (17, 113, 149, TO_DATE('02-05-2021',
'DD-MM-YYYY'), 4000);
INSERT INTO REZERVARE VALUES (17, 145, 164, TO_DATE('01-09-2021',
'DD-MM-YYYY'), 1800);

INSERT INTO REZERVARE VALUES (18, 128, 166, TO_DATE('23-11-2019',
'DD-MM-YYYY'), 2200);
INSERT INTO REZERVARE VALUES (18, 122, 160, TO_DATE('16-01-2021',
'DD-MM-YYYY'), 1000);

INSERT INTO REZERVARE VALUES (21, 103, 149, TO_DATE('23-11-2019',
'DD-MM-YYYY'), 8000);

INSERT INTO REZERVARE VALUES (19, 131, 166, TO_DATE('23-10-2019',
'DD-MM-YYYY'), 2200);
INSERT INTO REZERVARE VALUES (19, 106, 149, TO_DATE('24-08-2020',
'DD-MM-YYYY'), 2000);
INSERT INTO REZERVARE VALUES (19, 123, 158, TO_DATE('27-07-2021',
'DD-MM-YYYY'), 1800);
```

Database Explorer

```

> project 2 of 40
> project_sgbdb 2 of 41
> vlad 2 of 40
  
```

console [project_sgbdb] x

```

267 ✓ INSERT INTO REZERVARE VALUES (16, 136, 161, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 1100);
268 ✓ INSERT INTO REZERVARE VALUES (16, 109, 153, TO_DATE('21-04-2020', 'DD-MM-YYYY'), 1800);
269 ✓
270 ✓
271 ✓ INSERT INTO REZERVARE VALUES (20, 104, 149, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 2300);
272 ✓ INSERT INTO REZERVARE VALUES (20, 109, 150, TO_DATE('05-06-2020', 'DD-MM-YYYY'), 4400);
273 ✓ INSERT INTO REZERVARE VALUES (20, 128, 166, TO_DATE('09-08-2020', 'DD-MM-YYYY'), 5005);
274 ✓ INSERT INTO REZERVARE VALUES (20, 107, 149, TO_DATE('19-03-2021', 'DD-MM-YYYY'), 6700);
275 ✓ INSERT INTO REZERVARE VALUES (20, 115, 154, TO_DATE('27-10-2021', 'DD-MM-YYYY'), 3200);
276 ✓
277 ✓ INSERT INTO REZERVARE VALUES (17, 143, 164, TO_DATE('23-12-2020', 'DD-MM-YYYY'), 4500);
278 ✓ INSERT INTO REZERVARE VALUES (17, 138, 163, TO_DATE('08-02-2021', 'DD-MM-YYYY'), 3020);
279 ✓ INSERT INTO REZERVARE VALUES (17, 113, 149, TO_DATE('02-05-2021', 'DD-MM-YYYY'), 4000);
280 ✓ INSERT INTO REZERVARE VALUES (17, 145, 164, TO_DATE('01-09-2021', 'DD-MM-YYYY'), 1800);
281 ✓
282 ✓ INSERT INTO REZERVARE VALUES (18, 128, 166, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 2200);
283 ✓ INSERT INTO REZERVARE VALUES (18, 122, 160, TO_DATE('16-01-2021', 'DD-MM-YYYY'), 1000);
284 ✓
285 ✓ INSERT INTO REZERVARE VALUES (21, 103, 149, TO_DATE('23-11-2019', 'DD-MM-YYYY'), 8000);
286 ✓
287 ✓ INSERT INTO REZERVARE VALUES (19, 131, 166, TO_DATE('23-10-2019', 'DD-MM-YYYY'), 2200);
288 ✓ INSERT INTO REZERVARE VALUES (19, 106, 149, TO_DATE('24-08-2020', 'DD-MM-YYYY'), 2000);
289 ✓ INSERT INTO REZERVARE VALUES (19, 123, 158, TO_DATE('27-07-2021', 'DD-MM-YYYY'), 1800);
  
```

Services

Tx [2024-01-01 21:24:57] 1 row affected in 5 ms

project_sgbdb console 11 s 29 ms

Database Consoles > project_sgbdb > console [project_sgbdb]

Introspecting schema C##PROJECT_SGBD at level 2 in incremental mode

4:11 CRLF UTF-8 4 spaces

5°C Mostly cloudy Search Taskbar Spotify VS Code

Database Explorer

```

> project 2 of 40
> project_sgbdb 2 of 41
  
```

console [project_sgbdb] x

```

1 ✓ SELECT * FROM ANEXA;
  
```

Services

Tx [2024-01-01 21:24:57] 29 rows

COD_ANEXA	COD_HOTEL	NUME	CAPACITATE
1	36	10 Restaurant	95
2	37	10 Piscina	<null>
3	38	10 Sala de fitness	35
4	39	10 Spa	20
5	40	10 Parcare	40
6	41	10 Gradina	<null>
7	42	10 Teren de sport	<null>
8	43	10 Sala de conferinte	80
9	44	10 Sala de jocuri	30
10	45	6 Restaurant	95
11	46	6 Piscina	<null>
12	47	6 Sala de fitness	35
13	48	6 Spa	25
14	49	6 Parcare	30

Database Consoles > project_sgbdb > console [project_sgbdb]

5°C Mostly clear Search Taskbar Spotify VS Code

Database Explorer Database console [project_sgbdb] x

```
1 ✓ SELECT * FROM ANGAJAT;
```

Services

Output C##PROJECT_SGBD.ANGAJAT x

	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	1990-05-27	Manager	32000	<null>
5	69	6	Radulescu	Luca	1990-05-27	Manager	31000	<null>
6	70	7	Marinescu	Andrei	1990-05-27	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	4780	67
13	77	10	Stanciu	Stefan	1991-02-25	Chelner	4900	67
14	78	10	Monteanu	Mihai	1995-08-09	Chelner	5200	67

Database Consoles > project_sgbdb > Database console [project_sgbdb]

1/1/2024 10:03 PM

Database Explorer Database console [project_sgbdb] x

```
1 ✓ SELECT * FROM APROVIZIONARE;
```

Services

Output C##PROJECT_SGBD.APROVIZIONARE x

	COD_PARTENER	COD_HOTEL	DATA_APROVIZIONARE	COST
1	11	10	2023-04-23	1000
2	12	10	2023-05-28	1500
3	13	10	2023-05-05	2500
4	14	10	2023-03-01	4000
5	15	10	2023-01-30	1500
6	11	6	2023-04-21	2000
7	12	6	2023-05-24	2500
8	13	6	2023-05-09	3000
9	14	6	2022-03-01	5000
10	15	6	2023-02-25	2000
11	11	7	2023-04-19	900
12	12	7	2023-05-28	1100
13	13	7	2023-03-06	1800
14	15	7	2023-01-12	800

Database Consoles > project_sgbdb > Database console [project_sgbdb]

1/1/2024 10:03 PM

Database Explorer Database console [project_sgbdb] Tx: Manual

```
1 ✓ SELECT * FROM CAMERA;
```

Services

Output C##PROJECT_SGBD.CAMERA

	COD_CAMERA	COD_HOTEL	COD_TIP	NUMAR	ETAJ
1	103	10	27	80	8
2	104	10	27	72	7
3	105	10	28	55	5
4	106	10	28	63	6
5	107	10	28	61	6
6	108	10	25	20	2
7	109	10	25	28	2
8	110	10	26	29	2
9	111	10	26	22	2
10	112	10	24	18	1
11	113	10	24	14	1
12	114	10	23	13	1
13	115	10	23	12	1
14	116	6	27	66	6

Database Consoles > project_sgbdb > Database console [project_sgbdb]

1:22 CRLF UTF-8 4 spaces 10:03 PM 1/1/2024

Database Explorer Database console [project_sgbdb] Tx: Manual

```
1 ✓ SELECT * FROM CLIENT;
```

Services

Output C##PROJECT_SGBD.CLIENT

	COD_CLIENT	NUME	PRENUME	CNP
1	16	Popa	Florin	5014103009878
2	17	Stoica	Roxana	2980408230012
3	18	Enescu	Bogdan	1991016190779
4	19	Diaconu	Laura	6010101013335
5	20	Vasilescu	Andrei	1970305170569
6	21	Dinu	Ana	28806141110278
7	22	Marin	Cristian	1960103150375

Database Consoles > project_sgbdb > Database console [project_sgbdb]

1:22 CRLF UTF-8 4 spaces 10:03 PM 1/1/2024

Database Explorer Database Explorer [console [project_sgbdb]]

```
1 ✓ SELECT * FROM FACILITATE;
```

Services Services

Output C##PROJECT_SGBD.FACILITATE

	COD_FACILITATE	FACILITATE	PRET
1		29 Terasa	70
2		30 Vedere panoramică	65
3		31 Seif	60
4		32 Cada cu hidromasaj	75
5		33 Minibar	30
6		34 Pat suplimentar	45
7		35 Aer conditionat	40

Database Consoles > project_sgbdb > console [project_sgbdb]

1:25 CRLF UTF-8 4 spaces ⌂ 10:04 PM 1/1/2024

Database Explorer Database Explorer [console [project_sgbdb]]

```
1 ✓ SELECT * FROM HOTEL;
```

Services Services

Output C##PROJECT_SGBD.HOTEL

	COD_HOTEL	COD_LOCATIE	NUME	DATA_INFIINTARE	NR_CAMERE	STELE
1	6	1	Novotel	2019-10-23	70	5
2	7	2	Ramada Plaza	2017-08-21	40	4
3	8	3	Riviera	2016-05-02	30	4
4	9	4	Alpin	2012-09-27	40	5
5	10	5	Vega	2019-03-15	60	5

Database Consoles > project_sgbdb > console [project_sgbdb]

1:20 CRLF UTF-8 4 spaces ⌂ 10:04 PM 1/1/2024

sqlproject Version control

Database Explorer console [project_sgbdb]

project_sgbdb

- HOTEL
- LOCATIE
- OFERTA
- PACHET
- PARTENER
- REZERVARE
- TIP_CAMERA
- sequences 1
- PUBLIC

1 ✓ SELECT * FROM LOCATIE;

Services

Tx

Database project

- LOCATIE
- console

project_sgbdb

- console 76 ms

Output C##PROJECT_SGBD.LOCATIE

COD_LOCATIE	STRADA	NUMAR	ORAS	TARA
1	Calea Victoriei	150	Bucuresti	Romania
2	Calea Unirii	10	Craiova	Romania
3	Strada Dunarii	7	Dubova	Romania
4	Strada Poiana Soarelui	15	Brasov	Romania
5	Bulevardul Ferdinand	10	Constanta	Romania

Database Consoles > project_sgbdb > console [project_sgbdb]

1:22 CRLF UTF-8 ENG 10:04 PM 1/1/2024

5°C Mostly clear

The screenshot shows the Database Explorer pane with the 'LOCATIE' table selected. The table has columns: COD_LOCATIE, STRADA, NUMAR, ORAS, and TARA. The data consists of five rows: 1 (Calea Victoriei, 150, Bucuresti, Romania), 2 (Calea Unirii, 10, Craiova, Romania), 3 (Strada Dunarii, 7, Dubova, Romania), 4 (Strada Poiana Soarelui, 15, Brasov, Romania), and 5 (Bulevardul Ferdinand, 10, Constanta, Romania). The status bar at the bottom indicates the time as 1:22, date as 1/1/2024, and system status as 'CRLF' and 'ENG'.

sqlproject Version control

Database Explorer console [project_sgbdb]

project_sgbdb

- HOTEL
- LOCATIE
- OFERTA
- PACHET
- PARTENER
- REZERVARE
- TIP_CAMERA
- sequences 1
- PUBLIC

1 ✓ SELECT * FROM OFERTA;

Services

Tx

Database project

- LOCATIE
- console

project_sgbdb

- console 65 ms

Output C##PROJECT_SGBD.OFERTA

COD_TIP	COD_FACILITATE
1	23
2	24
3	24
4	24
5	25
6	25
7	25
8	25
9	25
10	26
11	26
12	26
13	26
14	27

Database Consoles > project_sgbdb > console [project_sgbdb]

1:21 CRLF UTF-8 ENG 10:04 PM 1/1/2024

5°C Mostly clear

The screenshot shows the Database Explorer pane with the 'OFERTA' table selected. The table has columns: COD_TIP and COD_FACILITATE. The data consists of 14 rows, ranging from (1, 23) to (14, 27). The status bar at the bottom indicates the time as 1:21, date as 1/1/2024, and system status as 'CRLF' and 'ENG'.

Database Explorer **console [project_sgbdb]**

```
1 ✓ SELECT * FROM PACHET;
```

project_sgbdb

- > HOTEL
- > LOCATIE
- > OFERTA
- > PACHET
- > PARTENER
- > REZERVARE
- > TIP_CAMERA
- > sequences 1
- > PUBLIC

Services

Tx

Database

- project
 - LOCATIE
 - console
- project_sgbdb
 - console 53 ms
 - console 53 ms

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

Database Consoles > project_sgbdb > console [project_sgbdb]

1:21 CRLF UTF-8 4 spaces ⌂ 10:04 PM 1/1/2024

5°C Mostly clear

Database Explorer **console [project_sgbdb]**

```
1 ✓ SELECT * FROM PARTENER;
```

project_sgbdb

- > HOTEL
- > LOCATIE
- > OFERTA
- > PACHET
- > PARTENER
- > REZERVARE
- > TIP_CAMERA
- > sequences 1
- > PUBLIC

Services

Tx

Database

- project
 - LOCATIE
 - console
- project_sgbdb
 - console 65 ms
 - console 65 ms

Output C#PROJECT_SGBD.PARTENER

	COD_PARTENER	NUME	TIP_PRODUSE
1	11	Cif	Curatenie
2	12	Lidl	Alimente
3	13	Somnia	Textile
4	14	Mobexpert	Mobilier
5	15	Techfit	Echipament sportiv

Database Consoles > project_sgbdb > console [project_sgbdb]

1:23 CRLF UTF-8 4 spaces ⌂ 10:04 PM 1/1/2024

5°C Mostly clear

sqlproject Version control

Database Explorer console [project_sgbdb]

project_sgbdb

- HOTEL
- LOCATIE
- OFERTA
- PACHET
- PARTENER
- REZERVARE
- TIP_CAMERA

sequences 1

PUBLIC

Services

Tx

Database

- project
- console
- project_sgbdb
- console 93 ms
- console 93 ms

Output C#PROJECT_SGBD.REZERVARE

	COD_CLIENT	COD_CAMERA	COD_PACHET	DATA_SOSIRE	PRET
1	16	136	161	2019-11-23	1100
2	16	189	153	2020-04-21	1800
3	20	104	149	2019-11-23	2300
4	20	109	150	2020-06-05	4400
5	20	128	166	2020-08-09	5005
6	28	187	149	2021-03-19	6700
7	20	115	154	2021-10-27	3200
8	17	143	164	2020-12-23	4500
9	17	138	163	2021-02-08	3020
10	17	113	149	2021-05-02	4000
11	17	145	164	2021-09-01	1800
12	18	128	166	2019-11-23	2200
13	18	122	168	2021-01-16	1000
14	21	183	149	2019-11-23	8000

Database Consoles > project_sgbdb > console [project_sgbdb]

1/1/2024 10:04 PM 5°C Mostly clear

sqlproject Version control

Database Explorer console [project_sgbdb]

project_sgbdb

- HOTEL
- LOCATIE
- OFERTA
- PACHET
- PARTENER
- REZERVARE
- TIP_CAMERA

sequences 1

PUBLIC

Services

Tx

Database

- project
- console
- project_sgbdb
- console 74 ms
- console 74 ms

Output C#PROJECT_SGBD.TIP_CAMERA

	COD_TIP	NUME	CAPACITATE	SUPRAFATA	PRET
1	23	Single	1	20	200
2	24	Dubla	2	28	350
3	25	Apartament	4	60	450
4	26	Suite	3	50	400
5	27	Penthouse	7	120	800
6	28	Royal	5	80	600

Database Consoles > project_sgbdb > console [project_sgbdb]

1/1/2024 10:05 PM 5°C Mostly clear

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

```
/* Pentru un numar de rezervari precizat la un anumit hotel, sa se afiseze numele clientilor care au acest numar de rezervari la hotelul respectiv, precum si datele la care au sosit si suma preturilor rezervarilor pentru fiecare. */
```

Rezolvare:

```
CREATE OR REPLACE 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_c
lienti(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;
/
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;
/

```

The screenshot shows the SSMS interface with the following details:

- Database Explorer:** Shows the database structure for 'project_sgbdb' with tables like LOCATIE, OFERTA, PACHET, PARTENER, REZERVARE, and TIP_CAMERA.
- Code Editor:** Displays the PL/SQL procedure 'afisare_rezervari_hotel_ex6'. The code is as follows:

```

-- Pentru un numar de rezervari precizat la un anumit hotel, sa se afiseze numele clientilor care au acest numar de rezervari
-- la hotelul respectiv, precum si datele la care au sosit si suma preturilor rezervarilor pentru fiecare.
|
CREATE OR REPLACE 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
    );
    -- tabelou 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();
    -- tabelou 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

```

Database Explorer

```

32 WHERE UPPER(nume) = UPPER(nume_hotel);
33
34 IF v_exista_hotel = 0 THEN
35   | RAISE exc_nume_hotel;
36 END IF;
37
38 SELECT cod_hotel INTO v_cod_hotel
39 FROM hotel
40 WHERE UPPER(nume) = UPPER(nume_hotel);
41
42 IF v_cod_hotel IS NULL THEN
43   | RAISE exc_nume_hotel;
44 END IF;
45
46 -- testez daca nr_rezervari este mai mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel hotel
47 SELECT COUNT(*) INTO v_nr_max_rezervari
48 FROM rezervare r JOIN camera c ON r.cod_camera = c.cod_camera
49 WHERE c.cod_hotel = v_cod_hotel;
50
51 IF nr_rezervari <= 0 OR nr_rezervari > v_nr_max_rezervari THEN
52   | RAISE exc_nr_rezervari;
53 END IF;
54
55 -- punem datele in lista_info_clienti
56 FOR client_info IN (
57   | SELECT c.cod_client as cod_client, c.nume || ' ' || c.prenume as nume_complet
58   | FROM rezervare r JOIN client c ON r.cod_client = c.cod_client
59   | JOIN camera cam ON r.cod_camera = cam.cod_camera
60   | JOIN hotel h ON cam.cod_hotel = h.cod_hotel
61   | WHERE h.cod_hotel = v_cod_hotel
62   | GROUP BY c.cod_client, c.nume, c.prenume

```

Database Consoles > project_sgbdb > console [project_sgbdb]

3:1 CRLF UTF-8 4 spaces ⌂ ENG 12:02 AM 1/3/2024

Database Explorer

```

63 HAVING COUNT(*) = nr_rezervari
64 ) LOOP
65   lista_info_clienti.EXTEND;
66   lista_info_clienti(lista_info_clienti.COUNT).cod_client := client_info.cod_client;
67   lista_info_clienti(lista_info_clienti.COUNT).nume_client := client_info.nume_complet;
68   lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire := lista_date_sosire();
69   preturi_clienti_rez(client_info.cod_client) := 0;
70   FOR rezervare_info IN (SELECT data_sosire, pret
71     | FROM rezervare r JOIN camera c ON r.cod_camera = c.cod_camera
72     | WHERE c.cod_hotel = v_cod_hotel AND r.cod_client = client_info.cod_client) LOOP
73     lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire.EXTEND;
74     lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire(lista_info_clienti(lista_info_clienti.COUNT).lista_date_sosire.COUNT) := rezervare_info.pret;
75   END LOOP;
76 END LOOP;
77
78 -- daca nu exista clienti care sa aiba nr_rezervari rezervari la hotelul cu numele nume_hotel
79 -- atunci dam raise la no_data_found
80 IF lista_info_clienti.COUNT = 0 THEN
81   | RAISE NO_DATA_FOUND;
82 ELSE
83   -- daca exista astfel de clienti, atunci afisam informatiile despre acestia
84   FOR i IN 1..lista_info_clienti.COUNT LOOP
85     DBMS_OUTPUT.PUT_LINE(A 'Nume client: ' || lista_info_clienti(i).nume_client);
86     DBMS_OUTPUT.PUT_LINE(A 'Lista date sosire: ');
87     FOR j IN 1..lista_info_clienti(i).lista_date_sosire.COUNT LOOP
88       DBMS_OUTPUT.PUT_LINE(A 'lista_info_clienti(i).lista_date_sosire(j)');
89     END LOOP;
90     DBMS_OUTPUT.PUT_LINE(A 'Suma preturi rezervari: ' || preturi_clienti_rez(lista_info_clienti(i).cod_client));
91     DBMS_OUTPUT.PUT_LINE(A '-----');
92   END LOOP;

```

Database Consoles > project_sgbdb > console [project_sgbdb]

3:1 CRLF UTF-8 4 spaces ⌂ ENG 12:02 AM 1/3/2024

Database Explorer

```

83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112

```

Database Consoles > project_sgbdb > console [project_sgbdb]

2°C Clear

Database Explorer

```

97
98
99
100
101
102
103
104
105
106
107
108
109
110

```

Services

Tx: project

Output: C##PROJECT_SGBD.REZERVARE

```

END;
[2024-01-02 22:40:16] completed in 7 ms
Name client: Vasilescu Andrei
Lista date sosire:
23-NOV-19
19-MAR-21
05-JUN-20
27-OCT-21
Suma preturi rezervari: 16600

```

Database Consoles > project_sgbdb > console [project_sgbdb]

2°C Partly cloudy

sqlproject Version control

Database Explorer

project_sgbdb

- LOCATIE
- OFERTA
- PACHET
 - columns 4
 - COD_PACHE
 - COD_HOTEL
 - BENEFICIU
 - PRET_NUMB
 - keys 1
 - foreign keys 1
 - indexes 1
- PARTENER
- REZERVARE
 - columns 5
 - COD_CLIENT

Playground

console [project_sgbdb] Tx: Manual ▾

```
97 WHEN exc_name_hotel THEN
98     DBMS_OUTPUT.PUT_LINE( 'Nu exista hotelul cu numele ' || nume_hotel || '!'); 
99 WHEN NO_DATA_FOUND THEN
100    DBMS_OUTPUT.PUT_LINE( 'Nu exista clienti care sa aiba ' || nr_rezervari || ' rezervari la hotelul cu numele ' || nume_hotel || '!'); 
101 WHEN OTHERS THEN
102    DBMS_OUTPUT.PUT_LINE( 'Avem alta eroare!' ); 
103
104 END;
105
106
107 BEGIN
108     afisare_rezervari_hotel_ex6( nr_rezervari, nume_hotel );
109 END;
110
```

Services

Tx

- project
- LOCATIE
- console
- project_sgbdb
 - console 33 ms
 - console 33 ms
 - afisare_rezervari_hotel_ex6(nr_rezervari, nume_hotel);
 - vlad
 - console_2
 - console_2
 - afisare_rezervari_hotel_ex6(nr_rezervari, nume_hotel);
 - console
 - console
 - afisare_rezervari_hotel_ex6(nr_rezervari, nume_hotel);

Output C##PROJECT_SGBD.REZERVARE

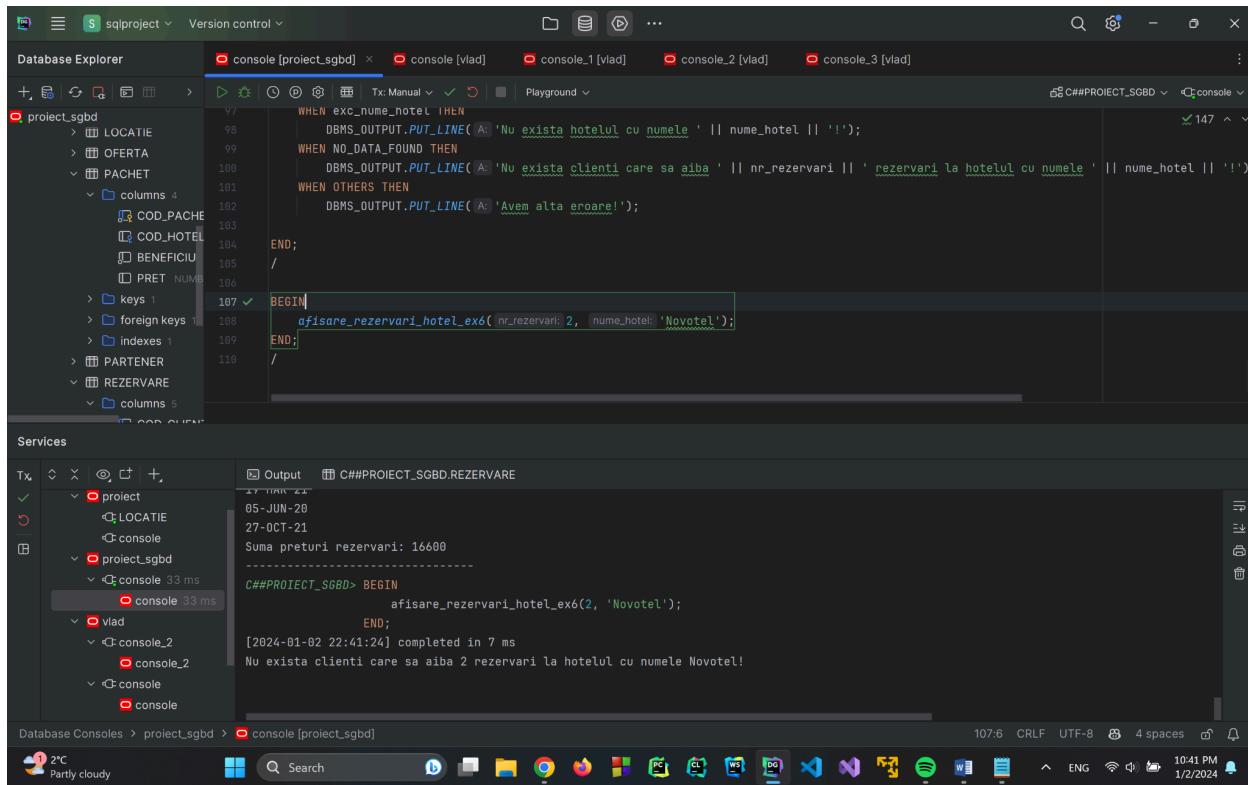
05-JUN-20
27-OCT-21
Suma preturi rezervari: 16600

C##PROJECT_SGBD> BEGIN
afisare_rezervari_hotel_ex6(2, 'Novotel');
END;
[2024-01-02 22:41:24] completed in 7 ms
Nu exista clienti care sa aiba 2 rezervari la hotelul cu numele Novotel!

Database Consoles > project_sgbdb > console [project_sgbdb]

107:6 CRLF UTF-8 4 spaces

2°C Party cloudy



sqlproject Version control

Database Explorer

project_sgbdb

- LOCATIE
- OFERTA
- PACHET
 - columns 4
 - COD_PACHE
 - COD_HOTEL
 - BENEFICIU
 - PRET_NUMB
 - keys 1
 - foreign keys 1
 - indexes 1
- PARTENER
- REZERVARE
 - columns 5
 - COD_CLIENT

Playground

console [project_sgbdb] Tx: Manual ▾

```
97 WHEN exc_name_hotel THEN
98     DBMS_OUTPUT.PUT_LINE( 'Nu exista hotelul cu numele ' || nume_hotel || '!'); 
99 WHEN NO_DATA_FOUND THEN
100    DBMS_OUTPUT.PUT_LINE( 'Nu exista clienti care sa aiba ' || nr_rezervari || ' rezervari la hotelul cu numele ' || nume_hotel || '!'); 
101 WHEN OTHERS THEN
102    DBMS_OUTPUT.PUT_LINE( 'Avem alta eroare!' ); 
103
104 END;
105
106
107 BEGIN
108     afisare_rezervari_hotel_ex6( nr_rezervari, nume_hotel );
109 END;
110
```

Services

Tx

- project
- LOCATIE
- console
- project_sgbdb
 - console 31 ms
 - afisare_rezervari_hotel_ex6(nr_rezervari, nume_hotel);
 - vlad
 - console_2
 - afisare_rezervari_hotel_ex6(nr_rezervari, nume_hotel);
 - console
 - afisare_rezervari_hotel_ex6(nr_rezervari, nume_hotel);

Output C##PROJECT_SGBD.REZERVARE

16-JAN-21
Suma preturi rezervari: 1000

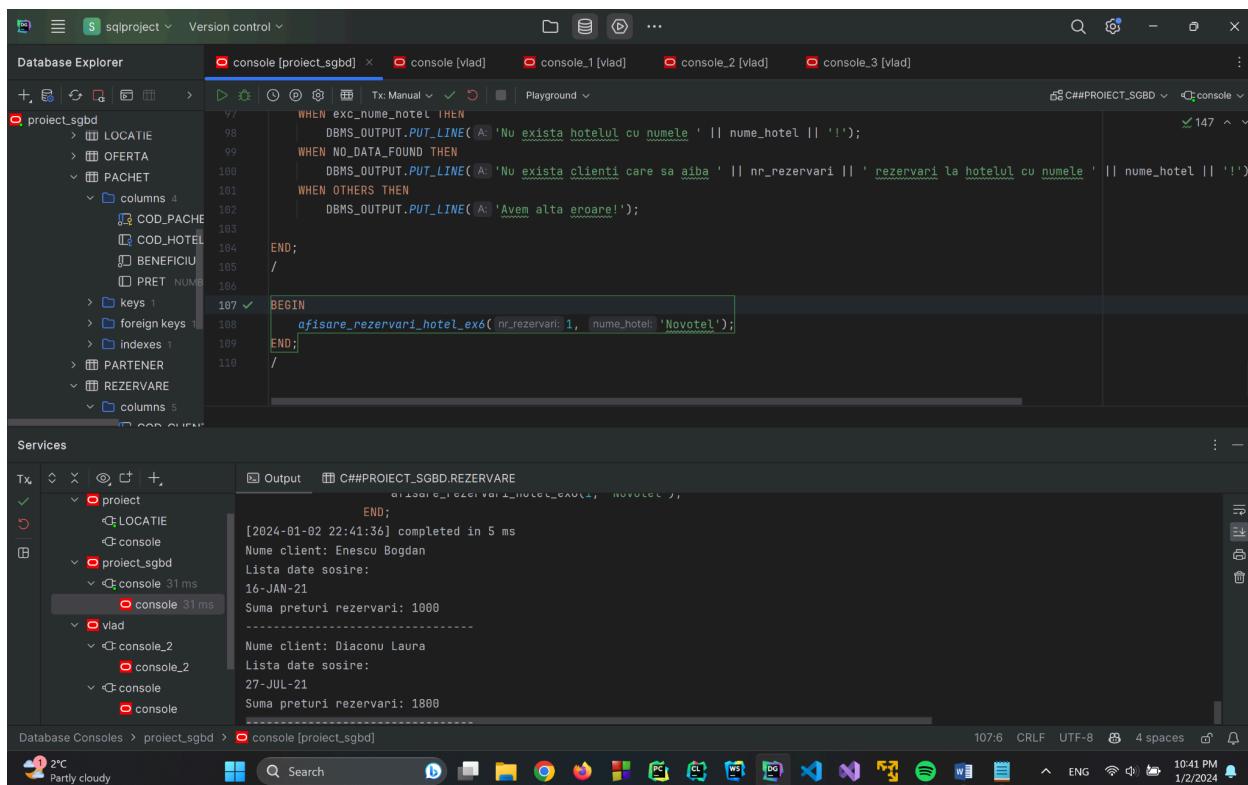
Nume client: Enescu Bogdan
Lista date sosire:
16-JAN-21
Suma preturi rezervari: 1800

Nume client: Diaconu Laura
Lista date sosire:
27-JUL-21
Suma preturi rezervari: 1800

Database Consoles > project_sgbdb > console [project_sgbdb]

107:6 CRLF UTF-8 4 spaces

2°C Party cloudy



The screenshot shows the SQL Server Management Studio interface. In the Database Explorer, a stored procedure named `afisare_rezervari_hotel_ex6` is selected. The code includes logic for handling different conditions (EXC_NAME_HOTEL, NO_DATA_FOUND, OTHERS) and a main loop that calls the stored procedure for each hotel. The Services pane displays the execution of this stored procedure for the hotel 'Novotel'. The output shows the date of arrival (27-JUL-21), the total reservation amount (Suma preturi rezervari: 1800), and a message indicating that the number of reservations should be less than or equal to the maximum number of reservations at that hotel.

```
WHEN exc_name_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;
/
BEGIN
    afisare_rezervari_hotel_ex6( nr_rezervari, nume_hotel );
END;
/
```

Output: C#PROIECT_SGBD.REZERVARE
nume_client: diaconu laura
Lista date sosire:
27-JUL-21
Suma preturi rezervari: 1800

C#PROIECT_SGBD> BEGIN
 afisare_rezervari_hotel_ex6(-1, 'Novotel');
END;
[2024-01-02 22:41:54] completed in 8 ms
Numarul de rezervari trebuie sa fie mai mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel hotel!

This screenshot shows the same environment as the previous one, but with a different input parameter for the stored procedure. Instead of '-1' (which resulted in an error), the parameter is now set to '20'. The output shows the same information as before, including the date of arrival (27-JUL-21), the total reservation amount (Suma preturi rezervari: 1800), and the message about the number of reservations.

```
WHEN exc_name_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;
/
BEGIN
    afisare_rezervari_hotel_ex6( nr_rezervari, nume_hotel );
END;
/
```

Output: C#PROIECT_SGBD.REZERVARE
nume_client: diaconu laura
Lista date sosire:
27-JUL-21
Suma preturi rezervari: 1800

C#PROIECT_SGBD> BEGIN
 afisare_rezervari_hotel_ex6(-1, 'Novotel');
END;
[2024-01-02 22:41:54] completed in 8 ms
Numarul de rezervari trebuie sa fie mai mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel hotel!
C#PROIECT_SGBD> BEGIN
 afisare_rezervari_hotel_ex6(20, 'Novotel');
END;
[2024-01-02 22:42:03] completed in 6 ms
Numarul de rezervari trebuie sa fie mai mare decat 0 si mai mic sau egal decat numarul maxim de rezervari din acel hotel!

sqlproject Version control

Database Explorer

```
185     DBMS_OUTPUT.PUT_LINE( A||'Numarul de rezervari trebuie sa fie mai mare decat & si mai mic sau egal decat numarul maxim de rezervari la hotelul cu numele '|| nume_hotel || '!');  
186 WHEN exc_name_hotel THEN  
187     DBMS_OUTPUT.PUT_LINE( A||'Nu exista hotelul cu numele '|| nume_hotel || '!');  
188 WHEN NO_DATA_FOUND THEN  
189     DBMS_OUTPUT.PUT_LINE( A||'Nu exista clienti care sa aiba '|| nr_rezervari || ' rezervari la hotelul cu numele '|| nume_hotel || '!');  
190 WHEN OTHERS THEN  
191     DBMS_OUTPUT.PUT_LINE( A||'Aveam alta eroare!');  
192  
193 END;  
194 /  
  
BEGIN  
    afisare_rezervari_hotel_ex6( nr_rezervari, nume_hotel );  
END;  
/
```

Services

Tx

- ✓ project
- ⌚ LOCATIE
- ⌚ console
- ⌚ project_sgbdb
- ⌚ console 45 ms
- ⌚ vlad

Output C##PROJECT_SGBD.REZERVARE

```
L##PROJECT_SGBDB> BEGIN  
    afisare_rezervari_hotel_ex6(2, 'Hotel1');  
END;  
[2024-01-02 22:57:34] completed in 10 ms  
Nu exista hotelul cu numele Hotel1!
```

Database Consoles > project_sgbdb > console [project_sgbdb]

2°C Partly cloudy

sqlproject Version control

Database Explorer

```
185     DBMS_OUTPUT.PUT_LINE( A||'Numarul de rezervari trebuie sa fie mai mare decat & si mai mic sau egal decat numarul maxim de rezervari la hotelul cu numele '|| nume_hotel || '!');  
186 WHEN exc_name_hotel THEN  
187     DBMS_OUTPUT.PUT_LINE( A||'Nu exista hotelul cu numele '|| nume_hotel || '!');  
188 WHEN NO_DATA_FOUND THEN  
189     DBMS_OUTPUT.PUT_LINE( A||'Nu exista clienti care sa aiba '|| nr_rezervari || ' rezervari la hotelul cu numele '|| nume_hotel || '!');  
190 WHEN OTHERS THEN  
191     DBMS_OUTPUT.PUT_LINE( A||'Aveam alta eroare!');  
192  
193 END;  
194 /  
  
BEGIN  
    afisare_rezervari_hotel_ex6( nr_rezervari, nume_hotel );  
END;  
/
```

Services

Tx

- ✓ project
- ⌚ LOCATIE
- ⌚ console
- ⌚ project_sgbdb
- ⌚ console 84 ms
- ⌚ vlad

Output C##PROJECT_SGBD.REZERVARE

```
L##PROJECT_SGBDB> BEGIN  
    afisare_rezervari_hotel_ex6(5, 'Vega');  
END;  
[2024-01-02 22:58:30] completed in 30 ms  
Nu exista clienti care sa aiba 5 rezervari la hotelul cu numele Vega!
```

Database Consoles > project_sgbdb > console [project_sgbdb]

2°C Partly cloudy

7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.

```
/* Sa se gaseasca rezervarile care se incadreaza intr-o anumita categorie
de pret (precizata la apelarea subprogramului), dupa cum urmeaza:
1) pret < 2500 - rezervare ieftina
2) 2500 <= pret <= 5000 - rezervare medie
3) pret > 5000 - rezervare scumpa
De asemenea, pentru fiecare client care a facut o rezervare din
categoria data, sa se afiseze numele sau si de cate ori a facut rezervari
la hotelul care se regaseste printre rezervarile aflate, la care a avut
numar maxim de rezervari in total, precum si pretul total platit pentru
rezervarile la acel hotel. */
```

Rezolvare:

```
CREATE OR REPLACE 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_numa_client, v_nr_rezervari,
v_pret_total;
        CLOSE c_client_hotel;

        DBMS_OUTPUT.PUT_LINE(v_numa_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;
/
DECLARE
    v_categ VARCHAR2(10) := &p_categ;
BEGIN
    rezervari_categorie_pret(v_categ);
EXCEPTION

```

```

WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Exceptie: ' || SQLERRM);
END;
/

```

```

-- Sa se gaseasca rezervarile care se incadreaza intr-o anumita categorie de pret, dupa cum urmeaza:
-- 1) pret < 2500 - rezervare ieftina
-- 2) 2500 <= pret <= 5000 - rezervare medie
-- 3) pret > 5000 - rezervare scumpa
-- De asemenea, pentru fiecare client care a facut o rezervare din categoria data, sa se afiseze numele sau si de cate ori a facut rezervari la hotelul care se regaseste printre rezervarile aflate, la care a avut numar maxim de rezervari in total, precum si pretul total platit pentru rezervarile la acel hotel.

CREATE OR REPLACE 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.name || ' ' || 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.name, 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_max 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;

```

Database Explorer

```

31   TYPE tablou_client_hotel_max IS TABLE OF hotel.cod_hotel%TYPE INDEX BY PLS_INTEGER;
32   client_hotel_max tablou_client_hotel_max;
33
34   BEGIN
35     IF LOWER(categ_rezervare) = 'ieftina' THEN
36       OPEN rezervari_categ FOR
37         SELECT r.cod_client, h.cod_hotel, r.pret
38         FROM rezervare r JOIN camera cam ON r.cod_camera = cam.cod_camera
39                   JOIN hotel h ON cam.cod_hotel = h.cod_hotel
40                   WHERE r.pret < 2500;
41     ELSIF LOWER(categ_rezervare) = 'medie' THEN
42       OPEN rezervari_categ FOR
43         SELECT r.cod_client, h.cod_hotel, r.pret
44         FROM rezervare r JOIN camera cam ON r.cod_camera = cam.cod_camera
45                   JOIN hotel h ON cam.cod_hotel = h.cod_hotel
46                   WHERE r.pret >= 2500 AND r.pret <= 5000;
47     ELSIF LOWER(categ_rezervare) = 'scumpa' THEN
48       OPEN rezervari_categ FOR
49         SELECT r.cod_client, h.cod_hotel, r.pret
50         FROM rezervare r JOIN camera cam ON r.cod_camera = cam.cod_camera
51                   JOIN hotel h ON cam.cod_hotel = h.cod_hotel
52                   WHERE r.pret > 5000;
53     ELSE
54       RAISE_APPLICATION_ERROR(-20000, 'Categoria de rezervari nu este valida!');
55     END IF;
56
57     LOOP
58       FETCH rezervari_categ INTO v_cod_client, v_cod_hotel, v_pret;
59       EXIT WHEN rezervari_categ%NOTFOUND;
60
61       IF client_hotel_max.EXISTS(v_cod_client) THEN
62         SELECT COUNT(*) INTO v_nr_rezervari;

```

Database Consoles > project_sgbdb > console_1 [project_sgbdb]

Cloudy Search 103:2 CRLF UTF-8 4 spaces ⌂ ENG 3:27 AM 1/3/2024

Database Explorer

```

60       IF client_hotel_max.EXISTS(v_cod_client) THEN
61         SELECT COUNT(*) INTO v_nr_rezervari
62         FROM rezervare r JOIN camera cam ON (r.cod_camera = cam.cod_camera)
63                   JOIN hotel h ON (cam.cod_hotel = h.cod_hotel)
64                   WHERE r.cod_client = v_cod_client AND h.cod_hotel = client_hotel_max(v_cod_client);
65
66         SELECT COUNT(*) INTO v_nr_rezervari_nou
67         FROM rezervare r JOIN camera cam ON (r.cod_camera = cam.cod_camera)
68                   JOIN hotel h ON (cam.cod_hotel = h.cod_hotel)
69                   WHERE r.cod_client = v_cod_client AND h.cod_hotel = v_cod_hotel;
70
71         IF v_nr_rezervari < v_nr_rezervari_nou THEN
72           client_hotel_max(v_cod_client) := v_cod_hotel;
73         END IF;
74       ELSE
75         client_hotel_max(v_cod_client) := v_cod_hotel;
76       END IF;
77     END LOOP;
78
79     CLOSE rezervari_categ;
80
81     DBMS_OUTPUT.PUT_LINE( A'Clientii care au facut rezervari din categoria ' || categ_rezervare || ' au istoricul:');
82     FOR client IN (SELECT cod_client FROM client) LOOP
83       IF client_hotel_max.EXISTS(client.cod_client) THEN
84         OPEN c_client_hotel(client.cod_client, client_hotel_max(client.cod_client));
85         FETCH c_client_hotel INTO v_nume_client, v_nr_rezervari, v_pret_total;
86         CLOSE c_client_hotel;
87
88         DBMS_OUTPUT.PUT_LINE( A' v_nume_client || ' a facut ' || v_nr_rezervari || ' rezervari la hotelul ' || client_hotel_max(client.cod_client));
89       END IF;
90     END LOOP;

```

Database Consoles > project_sgbdb > console_1 [project_sgbdb]

Cloudy Search 103:2 CRLF UTF-8 4 spaces ⌂ ENG 3:28 AM 1/3/2024

Database Explorer

```

    CLOSE rezervari_categ;
    DBMS_OUTPUT.PUT_LINE(A||'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(A||v_nume_client || ' a facut ' || v_nr_rezervari || ' rezervari la hotelul ' || client.hotel_max(client.cod_client));
        END IF;
    END LOOP;
END;
/
DECLARE
    v_categ VARCHAR2(10) := &p_categ;
BEGIN
    rezervari_categorie_pret(categ_rezervare, v_categ);
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE(A||'Excepție: ' || SQLERRM);
END;
/

```

Database Consoles > project_sgbdb > console_1 [project_sgbdb]

Database Explorer

```

    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(A||v_nume_client || ' a facut ' || v_nr_rezervari || ' rezervari la hotelul ' || client.hotel_max(client.cod_client));
END IF;
END LOOP;
END;
/
DECLARE
    v_categ VARCHAR2(10) := &p_categ;
BEGIN
    rezervari_categorie_pret(categ_rezervare, v_categ);
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE(A||'Excepție: ' || SQLERRM);
END;
/

```

Services

Tx	Tx	Output
✓	console 33 ms	COUNT(*) : NUMBER
✗	<:console_1 32 ms	EXCEPTION
	<:console_1 32 ms	WHEN OTHERS THEN
		DBMS_OUTPUT.PUT_LINE('Excepție: ' SQLERRM);
		END;
		[2024-01-03 03:25:35] completed in 8 ms
		Excepție: ORA-00000: Categoria de rezervari nu este validă!

Database Consoles > project_sgbdb > console_1 [project_sgbdb]

Database Explorer

```

84      OPEN c_client_hotel(client.cod_client, client.hotel_max(client.cod_client));
85      FETCH c_client_hotel INTO v_nume_client, v_nr_rezervari, v_pret_total;
86      CLOSE c_client_hotel;
87
88      DBMS_OUTPUT.PUT_LINE( A_ v_nume_client || ' a facut ' || v_nr_rezervari || ' rezervari la hotelul ' || client.hotel_max(client.cod_client));
89
90      END IF;
91      END LOOP;
92
93      END;
94
95      /
96
97      DECLARE
98          v_categ VARCHAR2(10) := &p_categ;
99
100     BEGIN
101         rezervari_categorie_pret( categ_rezervare, v_categ);
102
103         EXCEPTION
104             WHEN OTHERS THEN
105                 DBMS_OUTPUT.PUT_LINE( A_ 'Exceptie: ' || SQLERRM);
106     END;

```

Services

Tx

- console 33 ms
- console_1 36 ms
- console_1 36 ms
- vlad
- console_2
- console_2
- console
- Console

Output COUNT(*):NUMBER

```

EXECUTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Exceptie: ' || SQLERRM);
END;

```

[2024-01-03 03:26:27] completed in 6 ms

Exceptie: ORA-20000: Categoria de rezervari nu este valida!

Database Consoles > project_sgbdb > console_1 [project_sgbdb]

Database Explorer

```

84      OPEN c_client_hotel(client.cod_client, client.hotel_max(client.cod_client));
85      FETCH c_client_hotel INTO v_nume_client, v_nr_rezervari, v_pret_total;
86      CLOSE c_client_hotel;
87
88      DBMS_OUTPUT.PUT_LINE( A_ v_nume_client || ' a facut ' || v_nr_rezervari || ' rezervari la hotelul ' || client.hotel_max(client.cod_client));
89
90      END IF;
91      END LOOP;
92
93      END;
94
95      /
96
97      DECLARE
98          v_categ VARCHAR2(10) := &p_categ;
99
100     BEGIN
101         rezervari_categorie_pret( categ_rezervare, v_categ);
102
103         EXCEPTION
104             WHEN OTHERS THEN
105                 DBMS_OUTPUT.PUT_LINE( A_ 'Exceptie: ' || SQLERRM);
106     END;

```

Parameters

#1 v_categ VARCHAR2(10) := 'medie'

Execute Close

Services

Tx

- console 33 ms
- console_1 63 ms
- console_1 63 ms
- vlad
- console_2
- console_2
- console
- Console

Output COUNT(*):NUMBER

```

DBMS_OUTPUT.PUT_LINE('Exceptie: ' || SQLERRM);
END;

```

[2024-01-03 03:26:39] completed in 12 ms

Clientii care au facut rezervari din categoria medie au istoricul:
 Stoica Roxana a facut 2 rezervari la hotelul 8 cu pretul total de 6300
 Vasilescu Andrei a facut 4 rezervari la hotelul 10 cu pretul total de 16600

Database Consoles > project_sgbdb > console_1 [project_sgbdb]

The screenshot shows the Oracle SQL Developer interface. In the Database Explorer, there is a procedure named `c_client_hotel` under the `project_sgbdb` schema. The procedure uses a cursor to fetch data from the `client` table and then prints the results to the output window. The Services tab shows multiple database connections (Tx) and their execution times. The Output tab displays the results of the procedure execution, which includes the names of clients and the number of reservations they have made.

```

84      OPEN c_client_hotel(client.cod_client, client.hotel_max(client.cod_client));
85      FETCH c_client_hotel INTO v_nume_client, v_nr_rezervari, v_pret_total;
86      CLOSE c_client_hotel;
87
88      DBMS_OUTPUT.PUT_LINE( A_<v_nume_client || ' a facut ' || v_nr_rezervari || ' rezervari la hotelul ' || client.hotel_max(client.cod_client));
89      END IF;
90      END LOOP;
91
92      END;
93      /
94
95  DECLARE
96      v_categ VARCHAR2(10) := &p_categ;
97  BEGIN
98      rezervari_categorie_pret( Categ_rezervare, v_categ);
99      EXCEPTION
100         WHEN OTHERS THEN
101             DBMS_OUTPUT.PUT_LINE( A_<'Excepție: ' || SQLERRM);
102     END;

```

Output:

```

COUNT(*)NUMBER
DBMS_OUTPUT.PUT_LINE('Excepție: ' || SQLERRM);
[2024-01-03 03:26:47] completed in 5 ms
Clientii care au facut rezervari din categoria medie au istoricul:
Stoica Roxana a facut 2 rezervari la hotelul 8 cu pretul total de 6300
Vasilescu Andrei a facut 4 rezervari la hotelul 10 cu pretul total de 16600

```

8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

```
/* Pentru hotelurile care au fost înființate după o data precizată și care
au fost fost aprovisionate de un anumit partener, să se determine lista
angajaților care lucrează în acele hoteluri și care au salariul mai mare
decât media salariailor de pe jobul lor. */
```

Rezolvare:

```
CREATE OR REPLACE TYPE lista_numere_ex8 IS VARRAY(30) OF VARCHAR2(30);

CREATE OR REPLACE FUNCTION lista_ang_ex8(data_param
hotel.data_infiintare%TYPE, partener_param partener.nume%TYPE)
RETURN lista_numere_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;
/

DECLARE
    TYPE lista_angajati IS VARRAY(30) OF VARCHAR2(30);
    solutie lista_nume_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;

```

```

        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 aprovisionare 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;

    ROLLBACK;
END;
/

```

The screenshot shows the SSMS interface with the following details:

- File Explorer:** Shows the project structure under 'sqlproject' with nodes like 'Scratches and Consoles', 'Database Consoles', and 'Console'.
- Object Explorer:** Shows the 'Console' node expanded, listing multiple database consoles (e.g., 'console_1 [project_sgbdb]', 'console_2 [project_sgbdb]', etc.) and a user node 'vlad'.
- Script Editor:** Displays the PL/SQL script provided in the code block.
- Output Tab:** Shows the results of the executed script. It lists three employees: Neagu Dana, Cojocaru Cosmina, and Iomita Irina. It also includes error messages indicating that the hotel with ID 99 does not exist and that the partner with ID 12 does not exist.
- Status Bar:** Shows system information like temperature (-5°C), battery level, and network status.

The screenshot shows the Oracle SQL Developer interface. In the Database Explorer, a procedure named `list_angajati` is selected from the `CLIENT` schema. The code implements a cursor for employees and prints their names to the console. The Services tab shows a transaction history with multiple sessions running, including `console_2` and `console_1`. The Output tab displays the results of the procedure execution, listing employee names: Popescu Alexandru, Radulescu Luca, Avram Cristian, Albu Victor, Stanescu Ana, Enache Corina, Constantinescu Diana, Pop Adrian, and Tonescu Dorin. The system status bar at the bottom indicates it's 8°C, mostly cloudy, and the time is 9:19 PM on 1/3/2024.

```
DBMS_OUTPUT.PUT_LINE( A||'Alta eroare: ' || SQLERRM);
RETURN NULL;
END;

DECLARE
    TYPE lista_angajati IS VARRAY(30) OF VARCHAR2(30);
    solutie lista_angajati;
BEGIN
    solutie := lista_angajati(data_param: TO_DATE('01-01-2010', 'DD-MM-YYYY'), partener_param: 'Lidl');
    IF solutie IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE( A||'Lista angajatilor: ');
        FOR i IN 1 .. solutie.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE( A||solutie(i));
        END LOOP;
    END IF;
END;
```

This screenshot shows the same environment after modifying the procedure. The modification changes the date parameter to '01-01-2025'. The Output tab now displays a message indicating no hotels were found: "Nu s-a inaintat niciun hotel dupa acea data!". The rest of the interface, including the Database Explorer, Services tab, and system status bar, remains identical to the first screenshot.

```
DBMS_OUTPUT.PUT_LINE( A||'Alta eroare: ' || SQLERRM);
RETURN NULL;
END;

DECLARE
    TYPE lista_angajati IS VARRAY(30) OF VARCHAR2(30);
    solutie lista_angajati;
BEGIN
    solutie := lista_angajati(data_param: TO_DATE('01-01-2025', 'DD-MM-YYYY'), partener_param: 'Lidl');
    IF solutie IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE( A||'Lista angajatilor: ');
        FOR i IN 1 .. solutie.COUNT LOOP
            DBMS_OUTPUT.PUT_LINE( A||solutie(i));
        END LOOP;
    END IF;
END;
```

sqlproject Version control

Database Explorer

```

project_sgbd
  > CAMERA
  > CLIENT
    > columns 4
      COD_CLIENTE NUMBER(10)
      NUME VARCHAR2(50)
      PRENUME VARCHAR2(50)
      CNP CHAR(13)
    > keys 1
    > indexes 1
  > FACILITATE
  > HOTEL
    > columns 6
      COD_HOTEL NUMBER(10)
      COD_LOCAT NUMBER(10)
      NUME VARCHAR2(50)
  > DATA INCR

```

Services

```

Tx: project_sgbd
  > console_2 37 ms
  > console_1 98 ms
  > PARTENER 3 s 933 ms
  > vlad
  > console ?

```

Output C##PROJECT_SGBD.ANGAJAT

```

DBMS_OUTPUT.PUT_LINE('Lista angajatilor: ');
FOR i IN 1 .. solutie.COUNT LOOP
  DBMS_OUTPUT.PUT_LINE(solutie(i));
END LOOP;
END;
[2024-01-03 21:20:28] completed in 7 ms
Partenerul introdus nu exista!

```

Database Consoles > project_sgbd > console_2 [project_sgbd]

8°C Mostly cloudy

sqlproject Version control

Database Explorer

```

project 2 of 40
project_sgbd 2 of 41
  > C##PROJECT_SGBD
    > tables 13
    > routines 3
    > sequences 1
    > collection types 1
    > PUBLIC
    > Server Objects
  > vlad 2 of 40

```

Services

```

Tx: project_sgbd
  > console_2 32 ms
  > console_1 98 ms
  > PARTENER 3 s 933 ms
  > vlad
  > console ?

```

Output C##PROJECT_SGBD.HOTEL

```

INSERT INTO hotel VALUES (99, 1, 'Hilton', TO_DATE('01-01-2025', 'DD-MM-YYYY'), 65, 5);
INSERT INTO aprovisionare VALUES (12, 99, TO_DATE('01-01-2025', 'DD-MM-YYYY'), 200);

rollback;

DECLARE
  TYPE lista_angajati IS VARRAY(30) OF VARCHAR2(30);
  solutie lista_nume_ex8;
BEGIN
  solutie := lista_ang_ex8(data_param: TO_DATE('31-12-2024', 'DD-MM-YYYY'), partener_param: '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;
[2024-01-03 21:27:14] completed in 9 ms
Nu exista hoteluri care sa satisfaca conditiile!

```

Database Consoles > project_sgbd > console_2 [project_sgbd]

8°C Mostly cloudy

sqlproject Version control

```

Files      console_7 [project_sgbdb]  console_4 [project_sgbdb]  console_5 [project_sgbdb]  console_3 [project_sgbdb]  console_2 [project_sgbdb]  C# PROJECT_SGBD  console_2
↓
sqlproject C:\Users\vcuplD
Scratches and Consoles
  Database Consoles
    project
      project_sgbdb
        console [project_sgbdb]
        console_1 [project_sgbdb]
        console_2 [project_sgbdb]
        console_3 [project_sgbdb]
        console_4 [project_sgbdb]
        console_5 [project_sgbdb]
        console_6 [project_sgbdb]
        console_7 [project_sgbdb]
        console_8 [project_sgbdb]
        console_9 [project_sgbdb]
  vlad
    console [vlad]
    console_1 [vlad]
    console_2 [vlad]
    console_3 [vlad]
  Extensions

1 -- Pentru hotelurile care au fost înființate după o date precizată și care au fost aprovisionate de un anumit partener, să se determine lista angajaților care lucrează în acele hoteluri și care au sărările mai mari decât media sărărilor de pe joburi.
2 -- sa se determine lista angajatilor care lucreaza in acele hoteluri si care au salariul mai mare decat media salarilor de pe joburi.
3
4 CREATE OR REPLACE TYPE lista_numere_ex8 IS VARRAY(30) OF VARCHAR2(30);
5
6 CREATE OR REPLACE FUNCTION lista_ang_ex8(data_param hotel.data_infiintare%TYPE, partener_param partener.numar%TYPE)
7   RETURN lista_numere_ex8 IS
8
9   CURSOR hoteluri_ex8 IS
10     SELECT h.cod_hotel AS cod_hotel_cursor,
11           CURSOR(SELECT a.nume || ' ' || a.prenume AS nume_angajat
12                  FROM angajat a
13                 WHERE a.cod_hotel = h.cod_hotel
14                   AND a.salariu > (SELECT AVG(ang.salariu)
15                                         FROM angajat ang
16                                         WHERE ang.job = a.job
17                                         GROUP BY ang.job) ) AS lista_angajati
18
19     FROM hotel h;
20   sol_lista_angajati lista_numere_ex8 := lista_numere_ex8();
21   v_cod_hotel hotel.cod_hotel%TYPE;
22   v_lista_angajati SYS_REFCURSOR;
23   TYPE tabel_lista_angajati IS TABLE OF VARCHAR2(30) INDEX BY BINARY_INTEGER;
24   tab_lista_angajati tabel_lista_angajati;
25
26   TYPE lista_coduri_hoteluri IS VARRAY(30) OF hotel.cod_hotel%TYPE;
27   lista_hoteluri_aprovisionate lista_coduri_hoteluri;
28
29   exc_data EXCEPTION;
30   PRAGMA EXCEPTION_INIT(exc_data, -20001);
31   exc_partener EXCEPTION;
32   PRAGMA EXCEPTION_INIT(exc_partener, -20002);

```

Database Consoles > project_sgbdb > console_2 [project_sgbdb]

99:8 CRLF UTF-8 4 spaces ⌂ ENG 10:28 PM 1/8/2024

sqlproject Version control

```

Files      console_7 [project_sgbdb]  console_4 [project_sgbdb]  console_5 [project_sgbdb]  console_3 [project_sgbdb]  console_2 [project_sgbdb]  C# PROJECT_SGBD  console_2
↓
sqlproject C:\Users\vcuplD
Scratches and Consoles
  Database Consoles
    project
      project_sgbdb
        console [project_sgbdb]
        console_1 [project_sgbdb]
        console_2 [project_sgbdb]
        console_3 [project_sgbdb]
        console_4 [project_sgbdb]
        console_5 [project_sgbdb]
        console_6 [project_sgbdb]
        console_7 [project_sgbdb]
        console_8 [project_sgbdb]
        console_9 [project_sgbdb]
  vlad
    console [vlad]
    console_1 [vlad]
    console_2 [vlad]
    console_3 [vlad]
  Extensions

35   data_max hotel.data_infiintare%TYPE;
36   v_partener_param NUMBER;
37
38 BEGIN
39   SELECT MAX(data_infiintare) INTO data_max FROM hotel;
40   -- verific daca exista data (a fi mai mic sau egal cu data maxima la care s-a infiintat un hotel)
41   IF data_param > data_max THEN
42     RAISE exc_data;
43   END IF;
44
45   -- verific daca exista partenerul
46   SELECT COUNT(*) INTO v_partener_param
47   FROM partener
48   WHERE UPPER(numar) = UPPER(partener_param);
49   IF v_partener_param = 0 THEN
50     RAISE exc_partener;
51   END IF;
52
53   -- retin toate hotelurile care au fost aprovisionate de partenerul respectiv
54   SELECT h.cod_hotel BULK COLLECT INTO lista_hoteluri_aprovisionate
55   FROM hotel h JOIN aprovisionare ap ON h.cod_hotel = ap.cod_hotel
56     JOIN partener p ON ap.cod_partener = p.cod_partener
57   WHERE UPPER(p.numar) = UPPER(partener_param) AND h.data_infiintare > data_param;
58
59   -- verific daca exista un hotel cu conditiile date
60   IF lista_hoteluri_aprovisionate.COUNT = 0 THEN
61     RAISE exc_no_data_found;
62   END IF;
63
64   OPEN hoteluri_ex8;
65   LOOP

```

Database Consoles > project_sgbdb > console_2 [project_sgbdb]

99:8 CRLF UTF-8 4 spaces ⌂ ENG 10:28 PM 1/8/2024

sqlproject Version control

```

Files      console_7 [project_sgbdb] console_4 [project_sgbdb] console_5 [project_sgbdb] console_3 [project_sgbdb] console_2 [project_sgbdb] ...
Scratches and Consoles
  Database Consoles
    project
      project_sgbdb
        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]
      console_1 [vlad]
      console_2 [vlad]
      console_3 [vlad]
  Extensions

65   LOOP
66     FETCH hoteluri_ex8 INTO v_cod_hotel, v_lista_angajati;
67     EXIT WHEN hoteluri_ex8%NOTFOUND;
68     FETCH v_lista_angajati BULK COLLECT INTO tab_lista_angajati;
69     -- verifica daca v_cod_hotel se afla in lista hotelurilor aflate mai sus
70     FOR i IN 1 .. lista_hoteluri_aprovizionate.COUNT LOOP
71       IF v_cod_hotel = lista_hoteluri_aprovizionate(i) THEN
72         FOR j IN 1 .. tab_lista_angajati.COUNT LOOP
73           sol_lista_angajati.EXTEND;
74           sol_lista_angajati(sol_lista_angajati.COUNT) := tab_lista_angajati(j);
75         END LOOP;
76       END IF;
77     END LOOP;
78   CLOSE hoteluri_ex8;
79
80   RETURN sol_lista_angajati;
81 EXCEPTION
82   WHEN exc_data THEN
83     DBMS_OUTPUT.PUT_LINE( 'Nu s-a inregistrat niciun hotel dupa aceea data!');
84     RETURN NULL;
85   WHEN exc_partener THEN
86     DBMS_OUTPUT.PUT_LINE( 'Partenerul introdus nu exista!');
87     RETURN NULL;
88   WHEN exc_no_data_found THEN
89     DBMS_OUTPUT.PUT_LINE( 'Nu exista hoteluri care sa satisfaca conditiile!');
90     RETURN NULL;
91   WHEN OTHERS THEN
92     DBMS_OUTPUT.PUT_LINE( 'Alta eroare: ' || SQLERRM);
93     RETURN NULL;
94   END;
95

```

Database Consoles > project_sgbdb > console_2 [project_sgbdb]

99:8 CRLF UTF-8 4 spaces ⌂ ENG 10:28 PM 1/8/2024

sqlproject Version control

```

Files      console_7 [project_sgbdb] console_4 [project_sgbdb] console_5 [project_sgbdb] console_3 [project_sgbdb] console_2 [project_sgbdb] ...
Scratches and Consoles
  Database Consoles
    project
      project_sgbdb
        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]
      console_1 [vlad]
      console_2 [vlad]
      console_3 [vlad]
  Extensions

98   DECLARE
99     TYPE lista_angajati IS VARRAY(30) OF VARCHAR(30);
100    solutie lista_numere_ex8;
101
102 BEGIN
103
104   solutie := lista_ang_ex8( data_param: TO_DATE('01-01-2010', 'DD-MM-YYYY'), partener_param: 'Lidl');
105   IF solutie IS NOT NULL THEN
106     DBMS_OUTPUT.PUT_LINE( 'Lista angajatilor: ');
107     FOR i IN 1 .. solutie.COUNT LOOP
108       DBMS_OUTPUT.PUT_LINE( solutie(i));
109     END LOOP;
110   END IF;
111
112   DBMS_OUTPUT.PUT_LINE( '-----');
113
114   solutie := lista_ang_ex8( data_param: TO_DATE('01-01-2025', 'DD-MM-YYYY'), partener_param: 'Lidl');
115   IF solutie IS NOT NULL THEN
116     DBMS_OUTPUT.PUT_LINE( 'Lista angajatilor: ');
117     FOR i IN 1 .. solutie.COUNT LOOP
118       DBMS_OUTPUT.PUT_LINE( solutie(i));
119     END LOOP;
120   END IF;
121
122   DBMS_OUTPUT.PUT_LINE( '-----');
123
124   solutie := lista_ang_ex8( data_param: TO_DATE('01-01-2010', 'DD-MM-YYYY'), partener_param: 'IKEA');
125   IF solutie IS NOT NULL THEN
126     DBMS_OUTPUT.PUT_LINE( 'Lista angajatilor: ');
127     FOR i IN 1 .. solutie.COUNT LOOP
128       DBMS_OUTPUT.PUT_LINE( solutie(i));

```

Database Consoles > project_sgbdb > console_2 [project_sgbdb]

99:8 CRLF UTF-8 4 spaces ⌂ ENG 10:28 PM 1/8/2024

```

110 DBMS_OUTPUT.PUT_LINE( A || 'Lista angajatilor: ' );
111 FOR i IN 1 .. solutie.COUNT LOOP
112   DBMS_OUTPUT.PUT_LINE( A || solutie(i));
113 END LOOP;
114
115 END IF;
116
117 DBMS_OUTPUT.PUT_LINE( A || '-----');
118
119 solutie := lista_ang_ex8( data_param: TO_DATE('01-01-2010', 'DD-MM-YYYY'), partener_param: 'IKEA' );
120
121 IF solutie IS NOT NULL THEN
122   DBMS_OUTPUT.PUT_LINE( A || 'Lista angajatilor: ' );
123   FOR i IN 1 .. solutie.COUNT LOOP
124     DBMS_OUTPUT.PUT_LINE( A || solutie(i));
125   END LOOP;
126
127 END IF;
128
129 DBMS_OUTPUT.PUT_LINE( A || '-----');
130
131 INSERT INTO hotel VALUES (99, 1, 'Hilton', TO_DATE('01-01-2010', 'DD-MM-YYYY'), 65, 5);
132 INSERT INTO aprovisionare VALUES (12, 99, TO_DATE('01-01-2015', 'DD-MM-YYYY'), 200);
133
134 solutie := lista_ang_ex8( data_param: TO_DATE('31-12-2024', 'DD-MM-YYYY'), partener_param: 'Cif' );
135
136 IF solutie IS NOT NULL THEN
137   DBMS_OUTPUT.PUT_LINE( A || 'Lista angajatilor: ' );
138   FOR i IN 1 .. solutie.COUNT LOOP
139     DBMS_OUTPUT.PUT_LINE( A || solutie(i));
140   END LOOP;
141
142 END IF;
143
144 ROLLBACK;
145
146 /

```

9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

```

/* Sa se gaseasca numele clientilor care au facut rezervari la toate
hotelurile cu piscina la care s-a cazat clientul cu cele mai multe
rezervari, inclusiv clientul respectiv (daca exista mai multi astfel de
clienti, se va considera cel cu numele mai mic lexicografic). Dintre
acestia, se vor afisa doar aceia care au platit, in total, peste o suma
data pe rezervarile facute in tot lantul hotelier. Se va obtine, de
asemenea, si numele clientului dintre cei gasiti care a facut cele mai
putine rezervari la hotelurile cu piscina. */

```

Rezolvare:

```

CREATE OR REPLACE 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;
/

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;
/

```

The screenshot shows a SQL development environment with multiple database consoles running. The consoles are:

- console_7 [project_sgbd]
- console_4 [project_sgbd]
- console_5 [project_sgbd]
- console_3 [project_sgbd]

The code being run is a PL/SQL block:

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

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

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( cost_total_param: 1000, nume_client_param: v_nume_client);
IF v_nume_client IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE( A'Clientul cu cele mai putine rezervari: ' || v_nume_client);
END IF;
ROLLBACK;
```

The Services pane shows the output of the queries:

- Popa Florin
- Stoica Roxana
- Clientul cu cele mai putine rezervari: Dinu Ana
- Vasilescu Andrei
- Clientul cu cele mai putine rezervari: Vasilescu Andrei
- Dinu Ana
- Popescu Ion
- Vasilescu Andrei
- Diaconu Laura
- Popa Florin
- Stoica Roxana
- Prea multi clienti cu nr minim!

System status at the bottom: 130:8 CRLF UTF-8 4 spaces 10:34 PM 1/8/2024

The screenshot shows a SQL development environment with one active database console:

Database Consoles > project_sgbd > console_3 [project_sgbd]

The code being run is:

```
DECLARE
    v_nume_client VARCHAR2(30);
BEGIN
    afisare_clienti_hoteluri_piscina( cost_total_param: 1000, nume_client_param: v_nume_client);
    IF v_nume_client IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE( A'Clientul cu cele mai putine rezervari: ' || v_nume_client);
    END IF;
END;
/
select * from client;
```

The Services pane shows the output of the query:

- [2024-01-04 19:35:40] completed in 14 ms
- Dinu Ana
- Vasilescu Andrei
- Clientul cu cele mai putine rezervari: Dinu Ana

System status at the bottom: 132:6 CRLF UTF-8 4 spaces 7:35 PM 1/4/2024

The screenshot shows a dark-themed IDE interface. In the top navigation bar, there are icons for file operations, a search bar, and tabs labeled "sqlproject" and "Version control". Below the navigation bar is a "Files" sidebar containing project files like "sqlproject", "Scratches and Consoles", "Database Consoles", and "Extensions". A "Database Consoles" section is expanded, showing "project", "console_1 [project_sgbdb]", "console_2 [project_sgbdb]", and "console_3 [project_sgbdb]". The "console_3" entry is selected and highlighted in red. The main workspace is a code editor with the title "DBMS_OUTPUT" showing the following PL/SQL code:

```
127 /
128
129
130    DECLARE
131        v_nume_client VARCHAR2(30);
132    BEGIN
133        afisare_clienti_hoteluri_piscina( cost_total_param: 100000, nume_client_param: v_nume_client);
134        IF v_nume_client IS NOT NULL THEN
135            DBMS_OUTPUT.PUT_LINE('Clientul cu cele mai putine rezervari: ' || v_nume_client);
136        END IF;
137    /
138
139    select * from client;
140
```

Below the code editor is a "Services" panel with a "Tx" tab showing database connections and a "Result" tab showing the output of the executed code. The output shows:

```
afisare_clienti_hoteluri_piscina(100000, 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;
[2024-01-04 19:35:48] completed in 9 ms
Nu exista clienti care sa satisfaca cerintele!
```

The bottom status bar displays "132:6 CRLF UTF-8 4 spaces" and the system tray shows the date and time as "1/4/2024 7:35 PM".

This screenshot shows a dark-themed IDE with a "Database Explorer" window open. The "Database Explorer" sidebar lists tables such as "OFERTA", "PACHET", "PARTENER", and "REZERVARE". Under "REZERVARE", there is a "columns" node. The main workspace is a code editor with the title "DBMS_OUTPUT" showing the following PL/SQL code:

```
125
126
127
128
129
130
131
132
133    DBMS_OUTPUT.PUT_LINE('Afisare: ' || v_nume_client);
134
135
136
137
138
139
140
```

The code continues with data insertion and a report 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);
ROLLBACK;

DECLARE
    v_nume_client VARCHAR2(30);
BEGIN
    afisare_clienti_hoteluri_piscina( cost_total_param: 1000, nume_client_param: 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;
END;
```

Below the code editor is a "Services" panel with a "Tx" tab showing database connections and a "Result" tab showing the output of the executed code. The output shows a list of clients:

```
Dinu Ana
Popescu Ion
Vasilescu Andrei
Diaconu Laura
Popa Florin
Stoica Roxana
Prea multi clienti cu nr minim!
```

The bottom status bar displays "133:8 CRLF UTF-8 4 spaces" and the system tray shows the date and time as "1/4/2024 7:38 PM".

sqlproject Version control

Files

- sqlproject C:\Users\vcuplD
- Scratches and Consoles
 - Database Consoles
 - project
 - project_sgbdb
 - console [project_sgbd]
 - console_1 [project_sgbd]
 - console_2 [project_sgbd]
 - console_3 [project_sgbd]
- vlad
- Extensions

DBMS_OUTPUT

```

1 -- Sa se gasesc numele clientilor care au facut rezervari la toate hotelurile cu piscina la care s-a cazat clientul
2 -- cu cele mai multe rezervari, inclusiv clientul respectiv (daca exista mai multi astfel de clienti, se va considera cel
3 -- cu numele mai mic lexicografic). Dintre acestia, se vor afisa doar aceia
4 -- care au platin, in total, peste o suma data pe rezervarile facute in tot lantul hoteller.
5 -- Se va obtine, de asemenea, si numele clientului dintr-unul dintre cei gasiti care a facut cele mai putine rezervari la hotelurile cu piscina.

CREATE OR REPLACE PROCEDURE afisare_clienti_hoteluri_piscina(cost_total_param IN NUMBER, nume_client_param OUT VARCHAR2) IS
  TYPE tablu_clienti_piscina IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
  clienti_piscina tablu_clienti_piscina := tablu_clienti_piscina();
  ...
  BEGIN
    afisare_clienti.hoteluri_piscina();
  END;

```

Database Consoles > project_sgbdb > console_3 [project_sgbd]

10°C Cloudy 16:22 CRLF UTF-8 4 spaces ⌂ ENG 7:35 PM 1/4/2024

sqlproject Version control

Files

- sqlproject C:\Users\vcuplD
- Scratches and Consoles
 - Database Consoles
 - project
 - project_sgbdb
 - console [project_sgbd]
 - console_1 [project_sgbd]
 - console_2 [project_sgbd]
 - console_3 [project_sgbd]
- vlad
- Extensions

DBMS_OUTPUT

```

31 SELECT rez_cod_client, nr_rezervari INTO v_client_max_p, v_val_client_max_p
32 FROM (
33   SELECT DISTINCT r.cod_client as rez_cod_client, c.nume, COUNT(*) AS nr_rezervari
34   FROM rezervare r JOIN client c ON r.cod_client = c.cod_client
35     JOIN camera cam ON r.cod_camera = cam.cod_camera
36     JOIN hotel h ON cam.cod_hotel = h.cod_hotel
37   GROUP BY r.cod_client, c.nume
38   ORDER BY nr_rezervari DESC, c.nume)
39 FETCH FIRST 1 ROWS ONLY;
40
41 -- retinem pentru fiecare client cate rezervari are la hotelurile cu piscina
42 FOR rezervare IN (
43   SELECT DISTINCT r.cod_client as rez_cod_client, COUNT(*) AS nr_rezervari
44   FROM rezervare r JOIN client c ON r.cod_client = c.cod_client
45     JOIN camera cam ON r.cod_camera = cam.cod_camera
46     JOIN hotel h ON cam.cod_hotel = h.cod_hotel
47     JOIN anexa a ON h.cod_hotel = a.cod_hotel
48   WHERE UPPER(a.nume) = 'PIISCINA'
49   GROUP BY r.cod_client
50   ORDER BY nr_rezervari DESC) LOOP
51   clienti_piscina(rezervare.rez_cod_client) := rezervare.nr_rezervari;
52 END LOOP;
53
54 -- acum selectam clientii ceruti
55 SELECT COUNT(DISTINCT h.cod_hotel) INTO v_nr_hoteluri_piscina
56 FROM rezervare r JOIN client c ON (r.cod_client = c.cod_client)
57   JOIN camera cam ON (r.cod_camera = cam.cod_camera)
58   JOIN hotel h ON (cam.cod_hotel = h.cod_hotel)
59   JOIN anexa a ON (h.cod_hotel = a.cod_hotel)
60 WHERE UPPER(a.nume) = 'PIISCINA' AND r.cod_client = v_client_max_p;

```

Database Consoles > project_sgbdb > console_3 [project_sgbd]

10°C Cloudy 16:22 CRLF UTF-8 4 spaces ⌂ ENG 7:35 PM 1/4/2024

sqlproject C:\Users\vcuplD Version control

Files ▾

Database Consoles

- project
- project_sgbdb
- console [project_sgbd]
- console_1 [project_sgbd]
- console_2 [project_sgbd]
- console_3 [project_sgbd]
- vlad
- Extensions

DBMS_OUTPUT

```

SELECT c.cod_client
BULK COLLECT INTO clienti_cerutii
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) = 'PIISCINA' 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.num) = 'PIISCINA' 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 cerutii
FOR i IN clienti_cerutii.FIRST .. clienti_cerutii.LAST LOOP
    SELECT c.nume || ' ' || c.prenume INTO v_nume_client_aux
    FROM client c
    WHERE c.cod_client = clienti_cerutii(i);
    DBMS_OUTPUT.PUT_LINE( A v_nume_client_aux);
END LOOP;

```

afisare_clienti_hoteluri_piscina()

Database Consoles > project_sgbdb > console_3 [project_sgbd]

10°C Cloudy 16:22 CRLF UTF-8 4 spaces ⌂ 7:35 PM 1/4/2024

sqlproject C:\Users\vcuplD Version control

Files ▾

Database Consoles

- project
- project_sgbdb
- console [project_sgbd]
- console_1 [project_sgbd]
- console_2 [project_sgbd]
- console_3 [project_sgbd]
- vlad
- Extensions

DBMS_OUTPUT

```

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

FOR i IN clienti_cerutii.FIRST .. clienti_cerutii.LAST LOOP
    IF clienti_piscina(clienti_cerutii(i)) = v_nr_rez_min THEN
        clienti_rez_min := clienti_piscina(clienti_cerutii(i));
        clienti_rez_min(clienti_rez_min.LAST) := clienti_cerutii(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( A 'Nu exista clienti care sa satisfaca cerintele!');

```

afisare_clienti_hoteluri_piscina()

Database Consoles > project_sgbdb > console_3 [project_sgbd]

10°C Cloudy 16:22 CRLF UTF-8 4 spaces ⌂ 7:35 PM 1/4/2024

```
112      --RAISE TOO_MANY_ROWS;
113      END IF;
114
115
116      EXCEPTION
117          WHEN exc_no_data_found THEN
118              nume_client_param := NULL;
119              DBMS_OUTPUT.PUT_LINE(A||'Nu exista clienti care sa satisfaca cerintele!');
120          WHEN exc_too_many_rows THEN
121              nume_client_param := NULL;
122              DBMS_OUTPUT.PUT_LINE(A||'Prea multi clienti cu nr minim!');
123          WHEN OTHERS THEN
124              nume_client_param := NULL;
125              DBMS_OUTPUT.PUT_LINE(A||'Eroare: ' || SQLERRM);
126      END;
127
128
129
130      DECLARE
131          v_nume_client VARCHAR2(30);
132      BEGIN
133          afisare_clienti_hoteluri_piscina(cost_total_param: 1000, nume_client_param: v_nume_client);
134          IF v_nume_client IS NOT NULL THEN
135              DBMS_OUTPUT.PUT_LINE(A||'Clientul cu cele mai putine rezervari: ' || v_nume_client);
136          END IF;
137
138          DBMS_OUTPUT.PUT_LINE(A||'-----');
139
140          afisare_clienti_hoteluri_piscina(cost_total_param: 10000, nume_client_param: v_nume_client);
141          IF v_nume_client IS NOT NULL THEN
142              DBMS_OUTPUT.PUT_LINE(A||'Clientul cu cele mai putine rezervari: ' || v_nume_client);
143      END;
144
145
146
147
148
149
150
151
152
153
154
155
156
```

```
127      /
128
129
130      DECLARE
131          v_nume_client VARCHAR2(30);
132      BEGIN
133          afisare_clienti_hoteluri_piscina(cost_total_param: 1000, nume_client_param: v_nume_client);
134          IF v_nume_client IS NOT NULL THEN
135              DBMS_OUTPUT.PUT_LINE(A||'Clientul cu cele mai putine rezervari: ' || v_nume_client);
136          END IF;
137
138          DBMS_OUTPUT.PUT_LINE(A||'-----');
139
140          afisare_clienti_hoteluri_piscina(cost_total_param: 10000, nume_client_param: v_nume_client);
141          IF v_nume_client IS NOT NULL THEN
142              DBMS_OUTPUT.PUT_LINE(A||'Clientul cu cele mai putine rezervari: ' || v_nume_client);
143          END IF;
144
145          DBMS_OUTPUT.PUT_LINE(A||'-----');
146
147
148          INSERT INTO client VALUES (1000, 'Popescu', 'Ion', '0722222222');
149          INSERT INTO rezervare VALUES (1000, 103, 140, TO_DATE('24-08-2020', 'DD-MM-YYYY'), 2000);
150          afisare_clienti_hoteluri_piscina(cost_total_param: 1000, nume_client_param: v_nume_client);
151          IF v_nume_client IS NOT NULL THEN
152              DBMS_OUTPUT.PUT_LINE(A||'Clientul cu cele mai putine rezervari: ' || v_nume_client);
153          END IF;
154          ROLLBACK;
155
156      END;
157
158
```

10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

```
/* Pentru deschiderea unui nou hotel, s-a hotarat ca aceasta sa fie aprobată doar dacă media incasarilor pe rezervari a primelor 2 hoteluri ca nr de rezervari să fie mai mare de 21000 de lei. Creati un trigger care să verifice aceasta condiție și să nu permită deschiderea (inserarea) unui nou hotel dacă condiția nu este înăpunctată. */
```

Rezolvare:

```
CREATE OR REPLACE TRIGGER trigger_insert_hotel
BEFORE INSERT ON hotel
DECLARE
    TYPE info_hotel_ex10 IS RECORD (
        nr_rezervari NUMBER,
        incasari NUMBER
    );
    TYPE info_hotel_ex10_tab IS TABLE OF info_hotel_ex10;
    info_hotel info_hotel_ex10_tab := info_hotel_ex10_tab();
BEGIN
    FOR v_info_hotel IN ( SELECT h.cod_hotel AS cod_hotel, COUNT(*) AS nr_rezervari, SUM(r.pret) AS incasari
                           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
                           ORDER BY SUM(r.pret) DESC
                           FETCH FIRST 2 ROWS ONLY
    )
    LOOP
        info_hotel.EXTEND;
        info_hotel(info_hotel.LAST).nr_rezervari := v_info_hotel.nr_rezervari;
        info_hotel(info_hotel.LAST).incasari := v_info_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;
END;
```

```
/
```

```
INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020',
'DD-MM-YYYY'), 44, 3);
```

```
INSERT INTO rezervare VALUES (19, 103, 149, TO_DATE('01-01-2020',
'DD-MM-YYYY'), 2000);
ROLLBACK;
```

```
22
23     LOOP
24         info_hotel.EXTEND;
25         info_hotel(info_hotel.LAST).nr_rezervari := v_info_hotel.nr_rezervari;
26         info_hotel(info_hotel.LAST).incasari := v_info_hotel.incasari;
27     END LOOP;
28     IF (info_hotel(1).incasari + info_hotel(2).incasari) / 2 < 21000 THEN
29         RAISE_APPLICATION_ERROR(-20000, 'Nu se poate deschide un nou hotel');
30     END IF;
31     /
32
33     INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020', 'DD-MM-YYYY'), 44, 3);
34
[72000][20000]
ORA-20000: Nu se poate deschide un nou hotel
ORA-06512: at "C##PROJECT_SGBD.TRIGGER_INSERT_HOTEL", line 22
ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.TRIGGER_INSERT_HOTEL'
Position: 12
```

Services

Tx	Output
✓ Database	C##PROJECT_SGBD.HOTEL [2024-01-04 23:36:48] completed in 20 ms C##PROJECT_SGBD> INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020', 'DD-MM-YYYY'), 44, 3) [2024-01-04 23:36:52] [72000][20000] [2024-01-04 23:36:52] ORA-20000: Nu se poate deschide un nou hotel [2024-01-04 23:36:52] ORA-06512: at "C##PROJECT_SGBD.TRIGGER_INSERT_HOTEL", line 22 [2024-01-04 23:36:52] ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.TRIGGER_INSERT_HOTEL' [2024-01-04 23:36:52] Position: 12

Database Explorer

```

23     info_hotel.EXTEND;
24     info_hotel(info_hotel.LAST).nr_rezervari := v_info_hotel.nr_rezervari;
25     info_hotel(info_hotel.LAST).incasari := v_info_hotel.incasari;
26   END LOOP;
27   IF (info_hotel(1).incasari + info_hotel(2).incasari) / 2 < 21000 THEN
28     RAISE_APPLICATION_ERROR(-20000, 'Nu se poate deschide un nou hotel');
29   END IF;
30 END;
31 /
32
33 ✓ INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020', 'DD-MM-YYYY'), 44, 3);
34
35 INSERT INTO rezervare VALUES (19, 103, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 2000);
36 ROLLBACK;
37
38 SELECT * FROM REZERVARE;
39 SELECT * FROM HOTEL;
40

```

Services

Tx | Database | project_sgbd | console_4 [32 ms] | console_3 [90 ms]

```

[2024-01-04 23:36:52] ORA-06512: at "C##PROJECT_SGBD.TRIGGER_INSERT_HOTEL", line 22
[2024-01-04 23:36:52] ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.TRIGGER_INSERT_HOTEL'
[2024-01-04 23:36:52] Position: 12
C##PROJECT_SGBD> INSERT INTO rezervare VALUES (19, 103, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 2000)
[2024-01-04 23:36:59] 1 row affected in 15 ms
C##PROJECT_SGBD> INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020', 'DD-MM-YYYY'), 44, 3)
[2024-01-04 23:37:05] 1 row affected in 7 ms

```

Database Consoles > project_sgbd > console_4 [project_sgbd]

33:21 CRLF UTF-8 4 spaces

Database Explorer

```

1 -- Pentru deschiderea unui nou hotel, s-a hotarat ca aceasta sa fie aprobată doar daca media incasarilor pe rezervari
2 -- a primelor 2 hoteluri ca nr de rezervari a fie mai mare de 21000 de lei.
3 -- Creati un trigger care sa verifice aceaste conditie si sa nu permita deschiderea unui nou hotel daca conditia nu este indeplinita.
4
5 ✓ CREATE OR REPLACE TRIGGER trigger_insert_hotel
6 BEFORE INSERT ON hotel
7 DECLARE
8   TYPE info_hotel_ex10 IS RECORD (
9     nr_rezervari NUMBER,
10    incasari NUMBER
11  );
12  TYPE info_hotel_ex10_tab IS TABLE OF info_hotel_ex10;
13  info_hotel info_hotel_ex10_tab := info_hotel_ex10_tab();
14 BEGIN
15   FOR v_info_hotel IN ( SELECT h.cod_hotel AS cod_hotel, COUNT(*) AS nr_rezervari, SUM(r.pret) AS incasari
16                         FROM rezervare r JOIN camera cam ON r.cod_camera = cam.cod_camera
17                               JOIN hotel h ON cam.cod_hotel = h.cod_hotel
18                         GROUP BY h.cod_hotel
19                         ORDER BY SUM(r.pret) DESC
20                         FETCH FIRST 2 ROWS ONLY
21   )
22   LOOP
23     info_hotel.EXTEND;
24     info_hotel(info_hotel.LAST).nr_rezervari := v_info_hotel.nr_rezervari;
25     info_hotel(info_hotel.LAST).incasari := v_info_hotel.incasari;
26   END LOOP;
27   IF (info_hotel(1).incasari + info_hotel(2).incasari) / 2 < 21000 THEN
28     RAISE_APPLICATION_ERROR(-20000, 'Nu se poate deschide un nou hotel');
29   END IF;
30 END;
31 /
32
33 ✓

```

Services

Tx | Database | project_sgbd | console_4 [32 ms] | console_3 [90 ms]

```

[2024-01-04 23:36:52] ORA-06512: at "C##PROJECT_SGBD.HOTEL", line 22
[2024-01-04 23:36:52] ORA-04088: error during execution of trigger 'C##PROJECT_SGBD.HOTEL'
[2024-01-04 23:36:52] Position: 12
C##PROJECT_SGBD> INSERT INTO rezervare VALUES (19, 103, 149, TO_DATE('01-01-2020', 'DD-MM-YYYY'), 2000)
[2024-01-04 23:36:59] 1 row affected in 15 ms
C##PROJECT_SGBD> INSERT INTO hotel VALUES (99, 1, 'Hotel1', TO_DATE('01-01-2020', 'DD-MM-YYYY'), 44, 3)
[2024-01-04 23:37:05] 1 row affected in 7 ms

```

Database Consoles > project_sgbd > console_4 [project_sgbd]

5:11 CRLF UTF-8 4 spaces