**-PROIECT -**

**SISTEME DE GESTIUNE A BAZELOR DE DATE**

**INFORMATICĂ AN II**

**FIRMĂ DE TURISM**

**STUDENT:**

**MATEI ALEXANDRU-CRISTIAN**

**GRUPA 243**

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

**Prezentare pe scurt a bazei de date**

Proiectul urmareste realizarea unei baze de date pentru o firma de turism care isi doreste promovarea unor pachete de vacanta in vederea rezervarii acestora de catre clientii companiei.

**Regulile de funcționare**

Evidenta rezervarilor facute de clienti este tinuta de angajatii companiei, ce ajuta la facilitarea procesului de organizare al calatoriei.

Totodata, un angajat poate face rezervari pentru unul sau mai multi clienti, iar un client este ajutat de unul sau mai multi angajati pentru a finaliza rezervarea.

Un angajat poate lucra intr-un singur sediu al firmei si apartine unui singur departament.

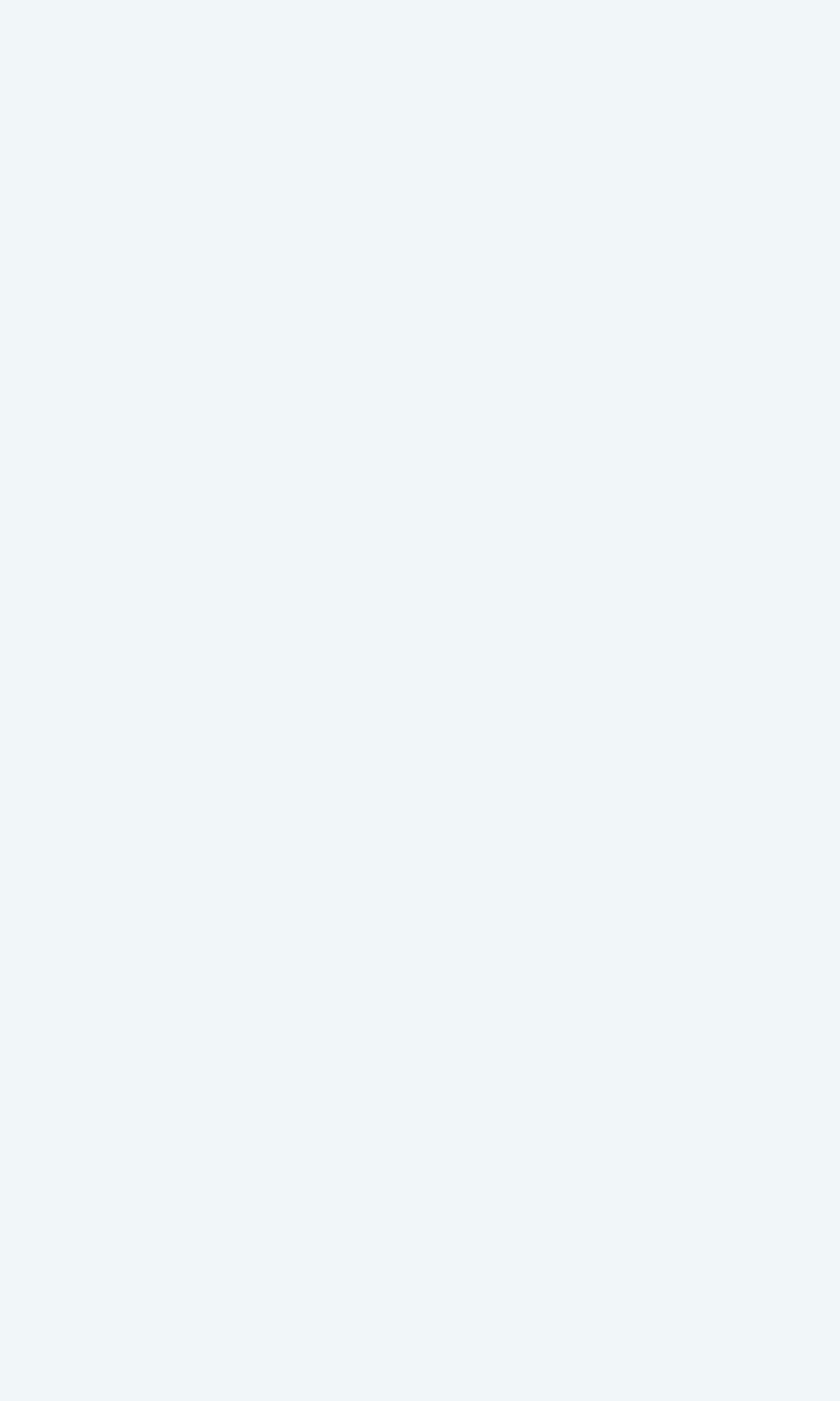
O rezervare este facuta pe baza unui singur pachet de vacanta, in timp ce un pachet poate fi prezent in una sau mai multe rezervari.

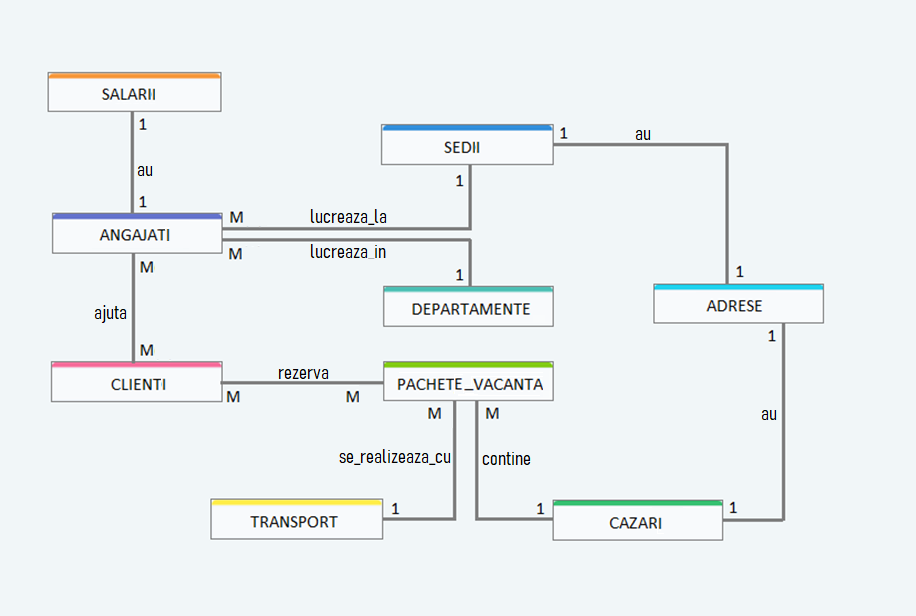
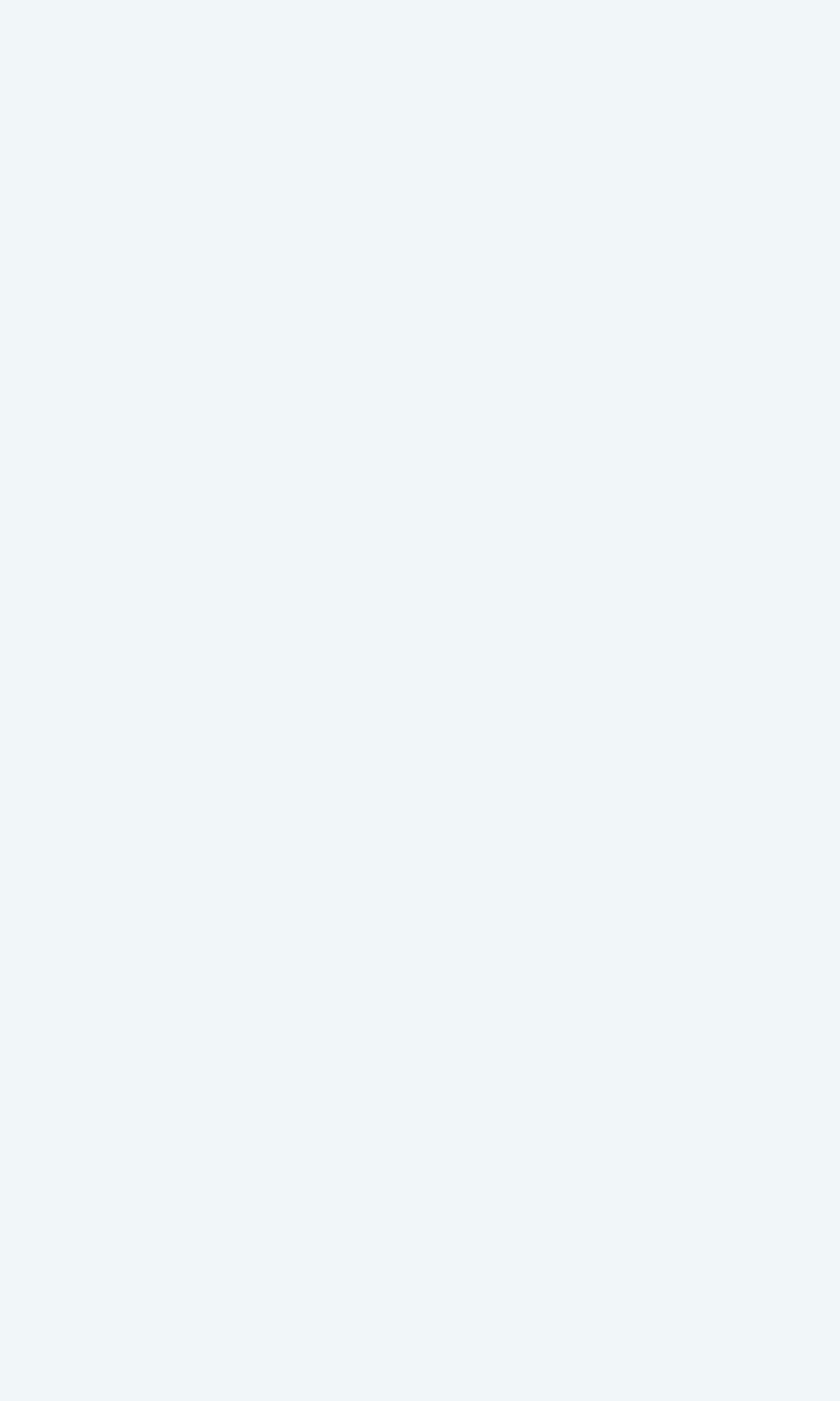
Aceeasi unitate de cazare poate fi regasita in unul sau mai multe pachete de vacanta, la fel ca tipul de transport.

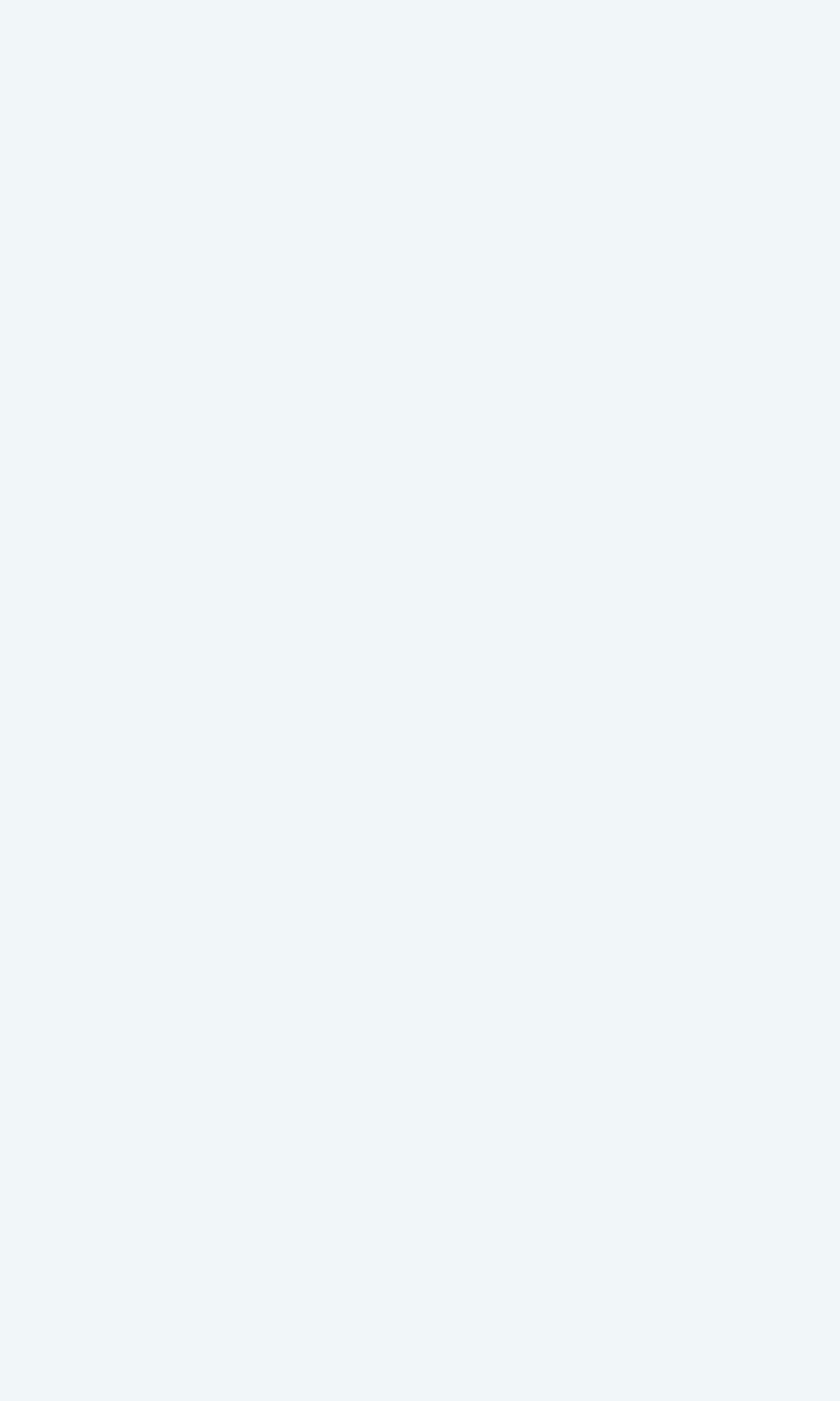
**Utilitatea bazei de date**

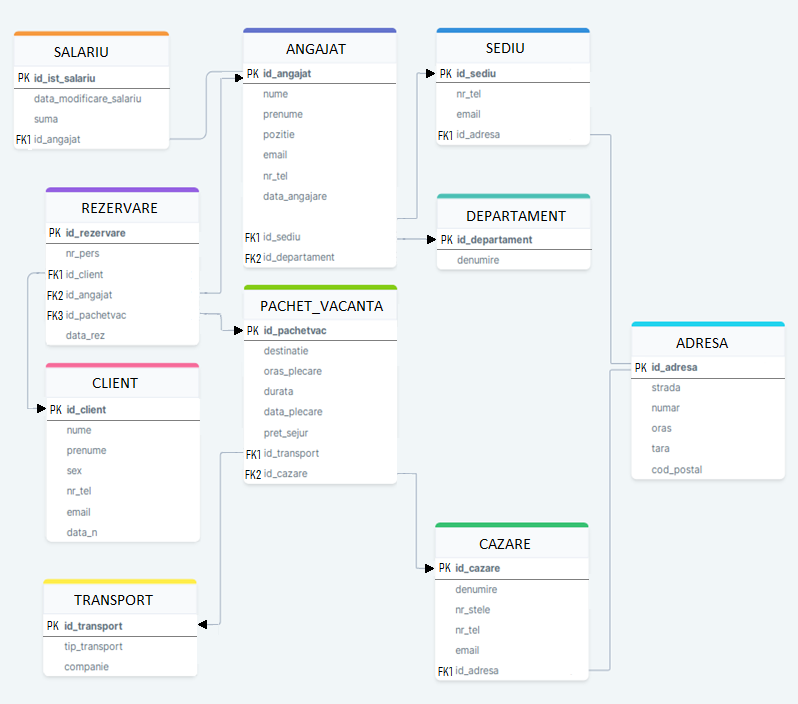
Din totalitatea atributelor existente in schema, se pot extrage informatii precum:

* Angajatul care a facilitat cele mai multe rezervari;
* Perioada din an cu cele mai multe rezervari;
* Topul destinatiilor in functie de diversi factori;
* Durata preferata a sejurului in functie de varsta clientilor;
* Sediile cu cele mai multe rezervari efectuate;
* Unitatile de cazare spre care au fost trimisi cei mai multi clienti.

1. **Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD / model de diagrama ERD; nu se va accepta alt format).**

****

1. **Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.**

****

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

create sequence SEQ\_CLIENT

increment by 1

start with 1

maxvalue 99999

minvalue 1

nocycle

nocache;

create sequence SEQ\_ANGAJAT

increment by 1

start with 1

maxvalue 9999

minvalue 1

nocycle

nocache;

create sequence SEQ\_SALARIU

increment by 1

start with 1

maxvalue 999

minvalue 1

nocycle

nocache;

create sequence SEQ\_REZERVARE

increment by 1

start with 1

maxvalue 999999

minvalue 1

nocycle

nocache;

create sequence SEQ\_PACHET\_VACANTA

increment by 1

start with 1

maxvalue 999

minvalue 1

nocycle

nocache;

create sequence SEQ\_SEDIU

increment by 1

start with 1

maxvalue 9999999

minvalue 1

nocycle

nocache;

create sequence SEQ\_ADRESA

increment by 1

start with 1

maxvalue 999999

minvalue 1

nocycle

nocache;

create sequence SEQ\_DEPARTAMENT

increment by 1

start with 1

maxvalue 999999

minvalue 1

nocycle

nocache;

create sequence SEQ\_CAZARE

increment by 1

start with 1

maxvalue 999999

minvalue 1

nocycle

nocache;

create sequence SEQ\_TRANSPORT

increment by 1

start with 1

maxvalue 9999999

minvalue 1

nocycle

nocache;

-- Tabel CLIENT

CREATE TABLE CLIENT (

id\_client NUMBER PRIMARY KEY,

nume VARCHAR2(30) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

sex VARCHAR2(8) NOT NULL,

nr\_tel CHAR(12) NOT NULL,

email VARCHAR2(50) NOT NULL,

data\_n DATE NOT NULL

);

-- Tabel ANGAJAT

CREATE TABLE ANGAJAT (

id\_angajat NUMBER PRIMARY KEY,

nume VARCHAR2(30) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

pozitie VARCHAR2(20) NOT NULL,

nr\_tel CHAR(12) NOT NULL,

email VARCHAR2(50) NOT NULL,

data\_angajare DATE NOT NULL,

id\_sediu NUMBER NOT NULL,

id\_departament NUMBER NOT NULL,

FOREIGN KEY (id\_sediu) REFERENCES SEDIU(id\_sediu),

FOREIGN KEY (id\_departament) REFERENCES DEPARTAMENT(id\_departament)

);

--Tabel SALARIU

CREATE TABLE SALARIU (

id\_ist\_salariu NUMBER PRIMARY KEY,

data\_modificare\_salariu DATE NOT NULL,

suma NUMBER NOT NULL,

id\_angajat NUMBER NOT NULL,

FOREIGN KEY (id\_angajat) REFERENCES ANGAJAT(id\_angajat)

);

-- Tabel REZERVARE

CREATE TABLE REZERVARE (

id\_rezervare NUMBER PRIMARY KEY,

nr\_pers NUMBER NOT NULL,

data\_rez DATE NOT NULL,

id\_client NUMBER NOT NULL,

id\_angajat NUMBER NOT NULL,

id\_pachetvac NUMBER NOT NULL,

FOREIGN KEY (id\_client) REFERENCES CLIENT(id\_client),

FOREIGN KEY (id\_angajat) REFERENCES ANGAJAT(id\_angajat),

FOREIGN KEY (id\_pachetvac) REFERENCES PACHET\_VACANTA(id\_pachetvac)

);

-- Tabel PACHET\_VACANTA

CREATE TABLE PACHET\_VACANTA (

id\_pachetvac NUMBER PRIMARY KEY,

destinatie VARCHAR2(20) NOT NULL,

oras\_plecare VARCHAR2(20) NOT NULL,

durata NUMBER NOT NULL,

data\_plecare DATE NOT NULL,

pret\_sejur NUMBER NOT NULL,

id\_transport NUMBER NOT NULL,

id\_cazare NUMBER NOT NULL,

FOREIGN KEY (id\_transport) REFERENCES TRANSPORT(id\_transport),

FOREIGN KEY (id\_cazare) REFERENCES CAZARE(id\_cazare)

);

-- Tabel SEDIU

CREATE TABLE SEDIU (

id\_sediu NUMBER PRIMARY KEY,

nr\_tel CHAR(12) NOT NULL,

email VARCHAR2(50) NOT NULL,

id\_adresa NUMBER NOT NULL,

FOREIGN KEY (id\_adresa) REFERENCES ADRESA(id\_adresa)

);

-- Tabel ADRESA

CREATE TABLE ADRESA (

id\_adresa NUMBER PRIMARY KEY,

strada VARCHAR2(20) NOT NULL,

numar NUMBER NOT NULL,

oras VARCHAR2(20) NOT NULL,

tara VARCHAR2(20) NOT NULL,

cod\_postal VARCHAR2(10) NOT NULL

);

-- Tabel DEPARTAMENT

CREATE TABLE DEPARTAMENT (

id\_departament NUMBER PRIMARY KEY,

denumire VARCHAR2(20) NOT NULL

);

-- Tabela CAZARE

CREATE TABLE CAZARE (

id\_cazare NUMBER PRIMARY KEY,

denumire VARCHAR2(20) NOT NULL,

nr\_stele NUMBER NOT NULL,

nr\_tel CHAR(12) NOT NULL,

email VARCHAR2(50) NOT NULL,

id\_adresa NUMBER NOT NULL,

FOREIGN KEY (id\_adresa) REFERENCES ADRESA(id\_adresa)

);

-- Tabel TRANSPORT

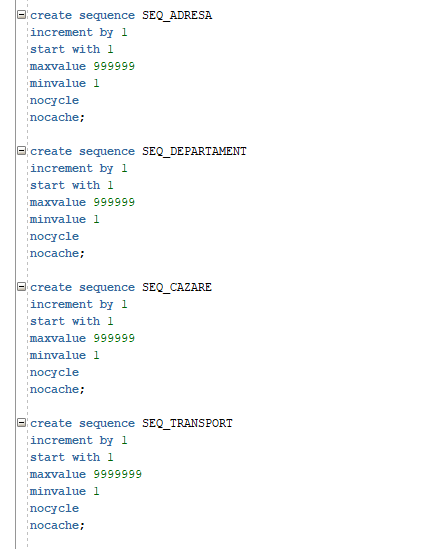
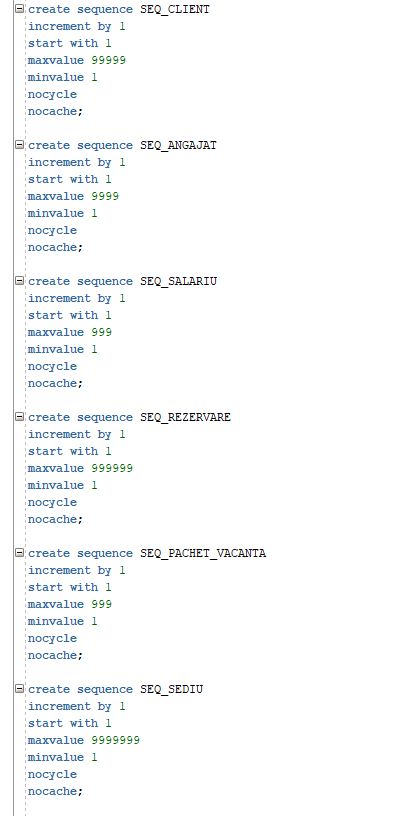
CREATE TABLE TRANSPORT (

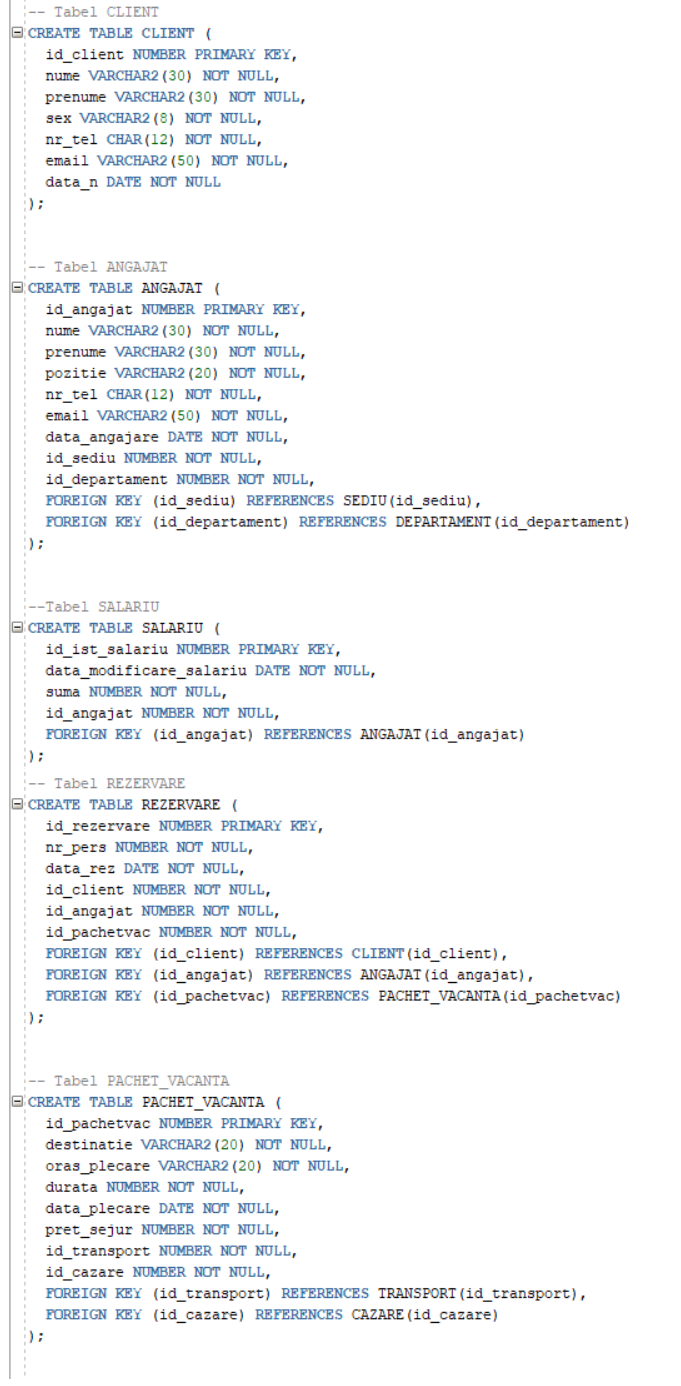
id\_transport NUMBER PRIMARY KEY,

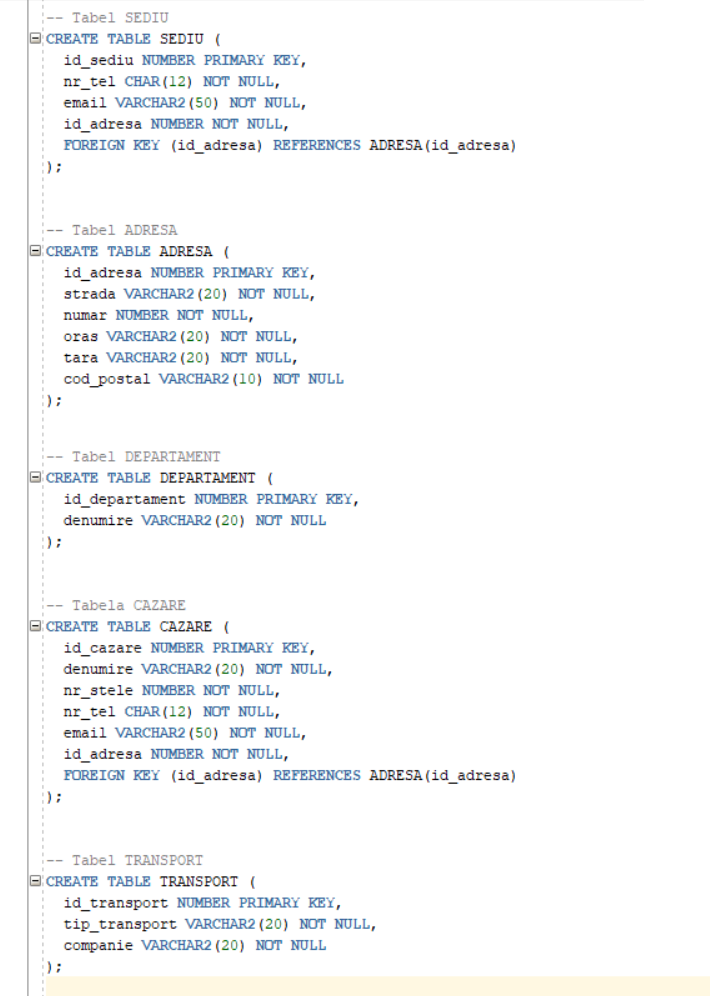
tip\_transport VARCHAR2(20) NOT NULL,

companie VARCHAR2(20) NOT NULL

);







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

-- Inserari TRANSPORT

INSERT INTO TRANSPORT (id\_transport, tip\_transport, companie) VALUES (SEQ\_TRANSPORT.NEXTVAL, 'Avion', 'BlueAir');

INSERT INTO TRANSPORT (id\_transport, tip\_transport, companie) VALUES (SEQ\_TRANSPORT.NEXTVAL, 'Autocar', 'EuroLines');

INSERT INTO TRANSPORT (id\_transport, tip\_transport, companie) VALUES (SEQ\_TRANSPORT.NEXTVAL, 'Tren', 'CFR');

INSERT INTO TRANSPORT (id\_transport, tip\_transport, companie) VALUES (SEQ\_TRANSPORT.NEXTVAL, 'Vapor', 'MSCruises');

INSERT INTO TRANSPORT (id\_transport, tip\_transport, companie) VALUES (SEQ\_TRANSPORT.NEXTVAL, 'Avion', 'Tarom');

-- Inserari DEPARTAMENT

INSERT INTO DEPARTAMENT (id\_departament, denumire) VALUES (SEQ\_DEPARTAMENT.NEXTVAL, 'Vânzari');

INSERT INTO DEPARTAMENT (id\_departament, denumire) VALUES (SEQ\_DEPARTAMENT.NEXTVAL, 'Marketing');

INSERT INTO DEPARTAMENT (id\_departament, denumire) VALUES (SEQ\_DEPARTAMENT.NEXTVAL, 'Rezervari');

INSERT INTO DEPARTAMENT (id\_departament, denumire) VALUES (SEQ\_DEPARTAMENT.NEXTVAL, 'IT');

INSERT INTO DEPARTAMENT (id\_departament, denumire) VALUES (SEQ\_DEPARTAMENT.NEXTVAL, 'Financiar');

-- Inserari ADRESA

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Strada Victoriei', 10, 'București', 'România', '010101');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Strada Avram Iancu', 25, 'Cluj-Napoca', 'România', '400001');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Strada Trandafirilor', 5, 'Timișoara', 'România', '300001');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Strada Salcamilor', 15, 'Iași', 'România', '700001');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Strada Republicii', 8, 'Brașov', 'România', '500001');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Main Street', 20, 'New York', 'SUA', '10001');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Rue de la Paix', 10, 'Paris', 'Franța', '75001');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Kurfürstendamm', 15, 'Berlin', 'Germania', '10707');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Calle Gran Via', 30, 'Madrid', 'Spania', '28013');

INSERT INTO ADRESA (id\_adresa, strada, numar, oras, tara, cod\_postal)

VALUES (SEQ\_ADRESA.NEXTVAL, 'Rua Augusta', 50, 'Lisabona', 'Portugalia', '110848');

--Inserari CAZARE

INSERT INTO CAZARE (id\_cazare, denumire, nr\_stele, nr\_tel, email, id\_adresa)

VALUES (SEQ\_CAZARE.NEXTVAL, 'Intercontinental', 4, '+40721234567', 'intercontinental@example.com', 6);

INSERT INTO CAZARE (id\_cazare, denumire, nr\_stele, nr\_tel, email, id\_adresa)

VALUES (SEQ\_CAZARE.NEXTVAL, 'Marriott', 3, '+40721234568', 'marriott@example.com', 7);

INSERT INTO CAZARE (id\_cazare, denumire, nr\_stele, nr\_tel, email, id\_adresa)

VALUES (SEQ\_CAZARE.NEXTVAL, 'Elite', 5, '+40721234569', 'elite@example.com', 8);

INSERT INTO CAZARE (id\_cazare, denumire, nr\_stele, nr\_tel, email, id\_adresa)

VALUES (SEQ\_CAZARE.NEXTVAL, 'SunsetInn', 5, '+40721234570', 'sunsetinn@example.com', 9);

INSERT INTO CAZARE (id\_cazare, denumire, nr\_stele, nr\_tel, email, id\_adresa)

VALUES (SEQ\_CAZARE.NEXTVAL, 'OceanView', 4, '+40721234571', 'oceanview@example.com', 10);

-- Inserari SEDIU

INSERT INTO SEDIU (id\_sediu, nr\_tel, email, id\_adresa)

VALUES (SEQ\_SEDIU.NEXTVAL, '+40721234597', 'sediu1@example.com', 1);

INSERT INTO SEDIU (id\_sediu, nr\_tel, email, id\_adresa)

VALUES (SEQ\_SEDIU.NEXTVAL, '+40721234598', 'sediu2@example.com', 2);

INSERT INTO SEDIU (id\_sediu, nr\_tel, email, id\_adresa)

VALUES (SEQ\_SEDIU.NEXTVAL, '+40721234599', 'sediu3@example.com', 3);

INSERT INTO SEDIU (id\_sediu, nr\_tel, email, id\_adresa)

VALUES (SEQ\_SEDIU.NEXTVAL, '+40721234590', 'sediu4@example.com', 4);

INSERT INTO SEDIU (id\_sediu, nr\_tel, email, id\_adresa)

VALUES (SEQ\_SEDIU.NEXTVAL, '+40721234591', 'sediu5@example.com', 5);

-- Inserari PACHET\_VACANTA

INSERT INTO Pachet\_vacanta (id\_pachetvac, destinatie, oras\_plecare, durata, data\_plecare, pret\_sejur, id\_transport, id\_cazare)

VALUES (SEQ\_PACHET\_VACANTA.NEXTVAL, 'New York', 'Bucuresti', 7, TO\_DATE('2023-06-01', 'YYYY-MM-DD'), 8000, 1, 1);

INSERT INTO Pachet\_vacanta (id\_pachetvac, destinatie, oras\_plecare, durata, data\_plecare, pret\_sejur, id\_transport, id\_cazare)

VALUES (SEQ\_PACHET\_VACANTA.NEXTVAL, 'Paris', 'Cluj-Napoca', 5, TO\_DATE('2023-07-10', 'YYYY-MM-DD'), 3500, 2, 2);

INSERT INTO Pachet\_vacanta (id\_pachetvac, destinatie, oras\_plecare, durata, data\_plecare, pret\_sejur, id\_transport, id\_cazare)

VALUES (SEQ\_PACHET\_VACANTA.NEXTVAL, 'Berlin', 'Timisoara', 3, TO\_DATE('2023-08-20', 'YYYY-MM-DD'), 3000, 3, 3);

INSERT INTO Pachet\_vacanta (id\_pachetvac, destinatie, oras\_plecare, durata, data\_plecare, pret\_sejur, id\_transport, id\_cazare)

VALUES (SEQ\_PACHET\_VACANTA.NEXTVAL, 'Madrid', 'Iasi', 6, TO\_DATE('2023-09-15', 'YYYY-MM-DD'), 4500, 5, 4);

INSERT INTO Pachet\_vacanta (id\_pachetvac, destinatie, oras\_plecare, durata, data\_plecare, pret\_sejur, id\_transport, id\_cazare)

VALUES (SEQ\_PACHET\_VACANTA.NEXTVAL, 'Lisabona', 'Constanta', 4, TO\_DATE('2023-10-05', 'YYYY-MM-DD'), 6000, 4, 5);

-- Inserari CLIENT

INSERT INTO Client (id\_client, nume, prenume, sex, nr\_tel, email, data\_n)

VALUES (SEQ\_CLIENT.NEXTVAL, 'Popescu', 'Ion', 'Masculin', '+40771234567', 'ion.popescu@example.com', TO\_DATE('1990-05-10', 'YYYY-MM-DD'));

INSERT INTO Client (id\_client, nume, prenume, sex, nr\_tel, email, data\_n)

VALUES (SEQ\_CLIENT.NEXTVAL, 'Ionescu', 'Maria', 'Feminin', '+40731234567', 'maria.ionescu@example.com', TO\_DATE('1985-12-20', 'YYYY-MM-DD'));

INSERT INTO Client (id\_client, nume, prenume, sex, nr\_tel, email, data\_n)

VALUES (SEQ\_CLIENT.NEXTVAL, 'Popa', 'Andrei', 'Masculin', '+40741234567', 'andrei.popa@example.com', TO\_DATE('1992-08-05', 'YYYY-MM-DD'));

INSERT INTO Client (id\_client, nume, prenume, sex, nr\_tel, email, data\_n)

VALUES (SEQ\_CLIENT.NEXTVAL, 'Georgescu', 'Elena', 'Feminin', '+40751234567', 'elena.georgescu@example.com', TO\_DATE('1988-03-15', 'YYYY-MM-DD'));

INSERT INTO Client (id\_client, nume, prenume, sex, nr\_tel, email, data\_n)

VALUES (SEQ\_CLIENT.NEXTVAL, 'Mihai', 'Alexandru', 'Masculin', '+40761234567', 'alexandru.mihai@example.com', TO\_DATE('1995-10-25', 'YYYY-MM-DD'));

-- Inserari ANGAJAT

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Dumitrescu', 'Maria', 'Manager de turism', '+40751234568', 'maria.dumitrescu@example.com', TO\_DATE('2017-04-05', 'YYYY-MM-DD'), 3, 3);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Constantin', 'Alexandru', 'Consultant turism', '+40723214567', 'alexandru.constantin@example.com', TO\_DATE('2022-05-20', 'YYYY-MM-DD'), 4, 1);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Popovici', 'Elena', 'Agent de turism', '+40729786543', 'elena.popovici@example.com', TO\_DATE('2021-06-12', 'YYYY-MM-DD'), 2, 3);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Stancu', 'Cristian', 'Agent de turism', '+40728675432', 'cristian.stancu@example.com', TO\_DATE('2020-07-09', 'YYYY-MM-DD'), 5, 3);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Stanescu', 'Andreea', 'Ghid turistic', '+40725456789', 'andreea.stanescu@example.com', TO\_DATE('2020-08-16', 'YYYY-MM-DD'), 3, 2);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Georgescu', 'Marius', 'Contabil', '+40721123456', 'marius.georgescu@example.com', TO\_DATE('2022-09-27', 'YYYY-MM-DD'), 1, 5);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Popov', 'Elena', 'Consultant turism', '+40728987654', 'elena.popov@example.com', TO\_DATE('2018-10-18', 'YYYY-MM-DD'), 2, 1);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Munteanu', 'Adrian', 'Agent de turism', '+40728765432', 'adrian.munteanu@example.com', TO\_DATE('2019-11-09', 'YYYY-MM-DD'), 1, 3);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_ANGAJAT.NEXTVAL, 'Diaconu', 'Roxana', 'Programator', '+40721134567', 'roxana.diaconu@example.com', TO\_DATE('2022-12-20', 'YYYY-MM-DD'), 3, 4);

INSERT INTO Angajat (id\_angajat, nume, prenume, pozitie, nr\_tel, email, data\_angajare, id\_sediu, id\_departament)

VALUES (SEQ\_angajat.NEXTVAL, 'Dumitru', 'Alexandru', 'Programator', '+40723219876', 'alexandru.dumitru@example.com', TO\_DATE('2021-08-10', 'YYYY-MM-DD'), 3, 4);

-- Inserari SALARIU

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2017-04-05', 'YYYY-MM-DD'), 5000, 1);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2022-05-20', 'YYYY-MM-DD'), 3500, 2);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2021-06-12', 'YYYY-MM-DD'), 2700, 3);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2020-07-09', 'YYYY-MM-DD'), 3000, 4);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2020-08-16', 'YYYY-MM-DD'), 2900, 5);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2022-09-27', 'YYYY-MM-DD'), 3000, 6);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2018-10-18', 'YYYY-MM-DD'), 3100, 7);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2019-11-09', 'YYYY-MM-DD'), 3200, 8);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2022-12-20', 'YYYY-MM-DD'), 3300, 9);

INSERT INTO Salariu (id\_ist\_salariu, data\_modificare\_salariu, suma, id\_angajat)

VALUES (SEQ\_SALARIU.NEXTVAL, TO\_DATE('2021-08-10', 'YYYY-MM-DD'), 3400, 10);

-- Inserari REZERVARE

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 2, TO\_DATE('2022-01-05', 'YYYY-MM-DD'), 1, 3, 1);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 1, TO\_DATE('2021-02-10', 'YYYY-MM-DD'), 2, 4, 2);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 3, TO\_DATE('2022-03-15', 'YYYY-MM-DD'), 3, 3, 3);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 2, TO\_DATE('2023-04-20', 'YYYY-MM-DD'), 4, 8, 4);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 1, TO\_DATE('2023-05-25', 'YYYY-MM-DD'), 5, 4, 5);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 4, TO\_DATE('2022-06-30', 'YYYY-MM-DD'), 2, 4, 2);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 3, TO\_DATE('2020-07-05', 'YYYY-MM-DD'), 3, 8, 5);

INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 2, TO\_DATE('2022-08-10', 'YYYY-MM-DD'), 5, 3, 1);

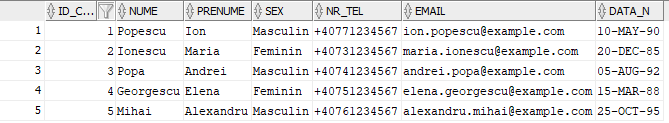
INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 1, TO\_DATE('2023-02-15', 'YYYY-MM-DD'), 1, 3, 4);

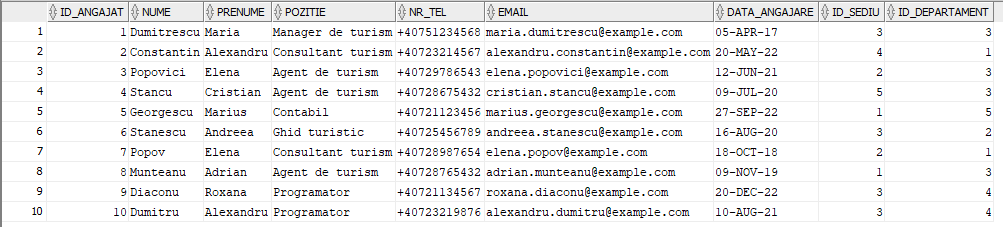
INSERT INTO Rezervare (id\_rezervare, nr\_pers, data\_rez, id\_client, id\_angajat, id\_pachetvac)

VALUES (SEQ\_REZERVARE.NEXTVAL, 2, TO\_DATE('2022-10-20', 'YYYY-MM-DD'), 4, 8, 1);

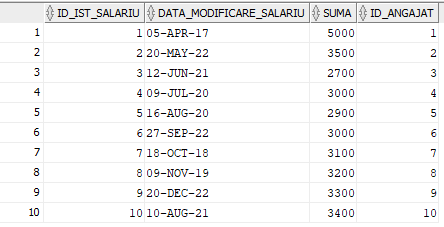
Tabelul CLIENT



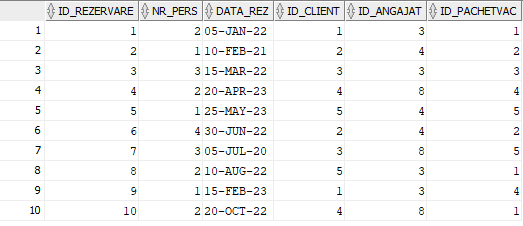
Tabelul ANGAJAT:



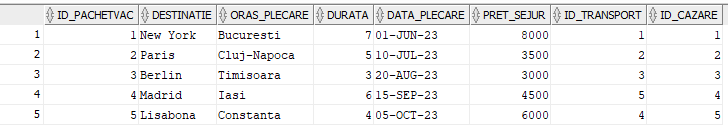
Tabelul SALARIU:



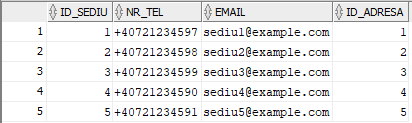
Tabelul REZERVARE:

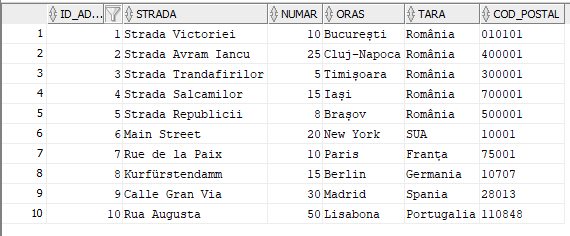


Tabelul PACHET\_VACANTA

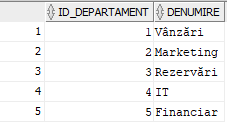


Tabelul SEDIU

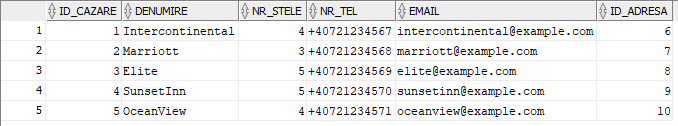


Tabelul ADRESA

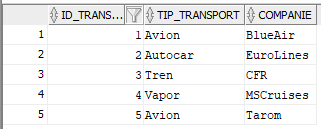
Tabelul DEPARTAMENT



Tabelul CAZARE



Tabelul TRANSPORT



1. **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.**

Sa se furnizeze informatii despre o rezervare specifica (numarul de persoane, data rezervarii). Sa se afiseze date relevante despre pachetul de vacanta din rezervarea respectiva si detaliile clientului care a efectuat rezervarea.

CREATE OR REPLACE PROCEDURE afisare\_informatii\_rezervare(

p\_id\_rezervare IN NUMBER

) AS

v\_data\_rez DATE;

v\_nr\_pers NUMBER;

-- Folosirea tablourilor imbricate (NESTED TABLES) pentru detaliile pachetului de vacanta

TYPE DetaliiPachetTable IS TABLE OF Pachet\_vacanta%ROWTYPE;

v\_detalii\_pachet DetaliiPachetTable;

-- Folosirea tablourilor indexate (INDEX BY TABLES) pentru detaliile rezervarii

TYPE DetaliiRezervareIndexByTable IS TABLE OF Rezervare%ROWTYPE INDEX BY PLS\_INTEGER;

v\_detalii\_rezervare DetaliiRezervareIndexByTable;

-- Folosirea vectorilor (VARRAYS) pentru detaliile clientului

TYPE DetaliiClientVarray IS VARRAY(5) OF Client%ROWTYPE;

v\_detalii\_client DetaliiClientVarray := DetaliiClientVarray();

BEGIN

-- Obtine informatii despre rezervare

SELECT data\_rez, nr\_pers

INTO v\_data\_rez, v\_nr\_pers

FROM Rezervare

WHERE id\_rezervare = p\_id\_rezervare;

-- Obtine informatii despre pachetul de vacanta din rezervare folosind tablourile imbricate (NESTED TABLES)

SELECT p.\*

BULK COLLECT INTO v\_detalii\_pachet

FROM Pachet\_vacanta p

WHERE p.id\_pachetvac = (SELECT id\_pachetvac FROM Rezervare WHERE id\_rezervare = p\_id\_rezervare);

-- Obtine informatii despre rezervare folosind tablourile indexate (INDEX BY TABLES)

SELECT r.\*

INTO v\_detalii\_rezervare(1)

FROM Rezervare r

WHERE r.id\_rezervare = p\_id\_rezervare;

-- Obtine informatii despre client folosind vectorii (VARRAYS)

SELECT c.\*

BULK COLLECT INTO v\_detalii\_client

FROM Client c

WHERE c.id\_client = (SELECT id\_client FROM Rezervare WHERE id\_rezervare = p\_id\_rezervare);

-- Afiseaza informatii despre rezervare folosind tablourile indexate (INDEX BY TABLES)

DBMS\_OUTPUT.PUT\_LINE('Detalii rezervare:');

DBMS\_OUTPUT.PUT\_LINE(' Numar persoane: ' || v\_detalii\_rezervare(1).nr\_pers);

DBMS\_OUTPUT.PUT\_LINE(' Data rezervare: ' || TO\_CHAR(v\_detalii\_rezervare(1).data\_rez, 'YYYY-MM-DD'));

-- Afiseaza informatii despre pachet folosind tablourile imbricate (NESTED TABLES)

FOR i IN 1..v\_detalii\_pachet.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Detalii pachet:');

DBMS\_OUTPUT.PUT\_LINE(' Destinatie: ' || v\_detalii\_pachet(i).destinatie);

DBMS\_OUTPUT.PUT\_LINE(' Oras plecare: ' || v\_detalii\_pachet(i).oras\_plecare);

DBMS\_OUTPUT.PUT\_LINE(' Durata: ' || v\_detalii\_pachet(i).durata || ' zile');

DBMS\_OUTPUT.PUT\_LINE(' Data plecare: ' || TO\_CHAR(v\_detalii\_pachet(i).data\_plecare, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE(' Pret sejur: ' || v\_detalii\_pachet(i).pret\_sejur);

END LOOP;

-- Afiseaza informatii despre client folosind vectorii (VARRAYS)

FOR i IN 1..v\_detalii\_client.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Detalii client care a efectuat rezervarea:');

DBMS\_OUTPUT.PUT\_LINE(' Nume: ' || v\_detalii\_client(i).nume);

DBMS\_OUTPUT.PUT\_LINE(' Prenume: ' || v\_detalii\_client(i).prenume);

DBMS\_OUTPUT.PUT\_LINE(' Sex: ' || v\_detalii\_client(i).sex);

DBMS\_OUTPUT.PUT\_LINE(' Numar de telefon: ' || v\_detalii\_client(i).nr\_tel);

DBMS\_OUTPUT.PUT\_LINE(' Email: ' || v\_detalii\_client(i).email);

DBMS\_OUTPUT.PUT\_LINE(' Data nastere: ' || TO\_CHAR(v\_detalii\_client(i).data\_n, 'YYYY-MM-DD'));

END LOOP;

END;

/

DECLARE

v\_id\_rezervare NUMBER := 3;

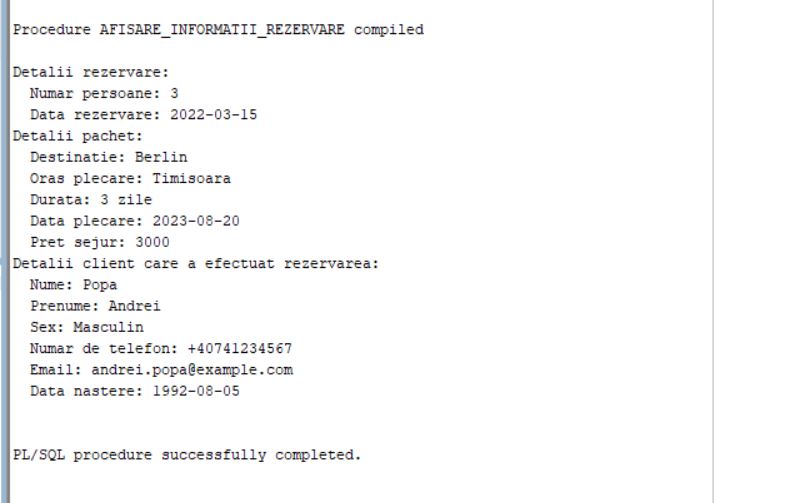
BEGIN

afisare\_informatii\_rezervare(p\_id\_rezervare => v\_id\_rezervare);

END;

/



****

1. **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.**

Creati un subprogram care va primi un interval de date și va returna o listă a rezervărilor efectuate în acea perioadă, împreună cu detalii suplimentare despre fiecare rezervare, cum ar fi numele și prenumele clientului, destinatia pachetului de vacanță și prețul sejurului.

CREATE OR REPLACE PROCEDURE detalii\_rezervari (

p\_data\_inceput IN DATE,

p\_data\_sfarsit IN DATE

) AS

-- Cursor clasic pentru a obține rezervările în intervalul de date specificat

CURSOR cur\_rezervari\_clasic IS

SELECT id\_rezervare, id\_client, data\_rez

FROM rezervare

WHERE data\_rez BETWEEN p\_data\_inceput AND p\_data\_sfarsit;

-- Cursor parametrizat pentru a obține detalii suplimentare despre fiecare rezervare

CURSOR cur\_detalii\_rezervare (p\_id\_rezervare NUMBER) IS

SELECT r.id\_rezervare, c.nume, c.prenume, p.destinatie, p.pret\_sejur, r.data\_rez

FROM rezervare r

JOIN client c ON r.id\_client = c.id\_client

JOIN pachet\_vacanta p ON r.id\_pachetvac = p.id\_pachetvac

WHERE r.id\_rezervare = p\_id\_rezervare;

-- Variabile pentru stocarea rezultatelor

v\_id\_rezervare NUMBER;

v\_nume\_client VARCHAR2(30);

v\_prenume\_client VARCHAR2(30);

v\_destinatie VARCHAR2(20);

v\_pret\_sejur NUMBER;

v\_data\_rez DATE;

BEGIN

-- Deschidem cursorul clasic și iterăm prin rezervări

FOR rez IN cur\_rezervari\_clasic LOOP

v\_id\_rezervare := rez.id\_rezervare;

-- Deschidem cursorul parametrizat și obținem detalii despre rezervare

FOR detalii IN cur\_detalii\_rezervare(v\_id\_rezervare) LOOP

v\_nume\_client := detalii.nume;

v\_prenume\_client := detalii.prenume;

v\_destinatie := detalii.destinatie;

v\_pret\_sejur := detalii.pret\_sejur;

v\_data\_rez := detalii.data\_rez;

-- Afișăm sau stocăm rezultatele cum este necesar

DBMS\_OUTPUT.PUT\_LINE('Rezervare ID: ' || v\_id\_rezervare);

DBMS\_OUTPUT.PUT\_LINE('Data rezervare: ' || v\_data\_rez);

DBMS\_OUTPUT.PUT\_LINE('Nume client: ' || v\_nume\_client || ' ' || v\_prenume\_client);

DBMS\_OUTPUT.PUT\_LINE('Destinatie: ' || v\_destinatie);

DBMS\_OUTPUT.PUT\_LINE('Pret sejur: ' || v\_pret\_sejur);

DBMS\_OUTPUT.PUT\_LINE('-------------------');

END LOOP;

END LOOP;

END detalii\_rezervari;

/

-- Declaram variabile pentru perioada de cautare

DECLARE

v\_data\_inceput DATE := TO\_DATE('2022-01-01', 'YYYY-MM-DD');

v\_data\_sfarsit DATE := TO\_DATE('2022-03-20', 'YYYY-MM-DD');

BEGIN

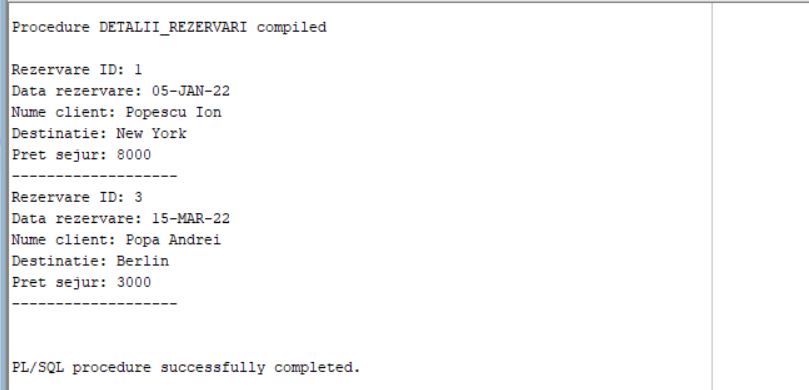
-- Apelăm subprogramul cu valorile definite

detalii\_rezervari(v\_data\_inceput, v\_data\_sfarsit);

END;

/





1. **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 un pachet de vacanta anume, sa se afiseze id-ul pachetului, destinatia, denumirea cazarii specifice pachetului de vacanta, numarul de stele cazare si adresa la care se afla cazarea respectiva (tara, oras, strada, numar). Daca nu exista un pachet cu id-ul respectiv se va afisa o exceptie. Daca se gaseste un pachet al carei cazare are numarul de stele <= 3 se va afisa o exceptie.

CREATE OR REPLACE FUNCTION detalii\_pachet\_vacanta (

id\_pachet\_param IN NUMBER

) RETURN VARCHAR2

IS

v\_cursor SYS\_REFCURSOR;

v\_id\_pachet NUMBER;

v\_destinatie VARCHAR2(20);

v\_denumire\_cazare VARCHAR2(20);

v\_nr\_stele\_cazare NUMBER;

v\_tara VARCHAR2(20);

v\_oras VARCHAR2(20);

v\_strada VARCHAR2(20);

v\_numar NUMBER;

NO\_ID\_FOUND EXCEPTION;

TOO\_FEW\_STARS EXCEPTION;

BEGIN

OPEN v\_cursor FOR

SELECT pv.id\_pachetvac, pv.destinatie, c.denumire, c.nr\_stele, a.tara, a.oras, a.strada, a.numar

FROM PACHET\_VACANTA pv

JOIN CAZARE c ON pv.id\_cazare = c.id\_cazare

JOIN ADRESA a ON c.id\_adresa = a.id\_adresa

WHERE pv.id\_pachetvac = id\_pachet\_param;

FETCH v\_cursor INTO v\_id\_pachet, v\_destinatie, v\_denumire\_cazare, v\_nr\_stele\_cazare, v\_tara, v\_oras, v\_strada, v\_numar;

IF v\_cursor%NOTFOUND THEN

CLOSE v\_cursor;

RAISE NO\_ID\_FOUND;

END IF;

IF v\_nr\_stele\_cazare <= 3 THEN

CLOSE v\_cursor;

RAISE TOO\_FEW\_STARS;

END IF;

CLOSE v\_cursor;

RETURN 'Detalii pachet de vacanta ID ' || v\_id\_pachet || ':' || CHR(10) ||

'Destinatie: ' || v\_destinatie || CHR(10) ||

'Cazare: ' || v\_denumire\_cazare || ' - ' || v\_nr\_stele\_cazare || ' stele' || CHR(10) ||

'Adresa: ' || v\_tara || ', ' || v\_oras || ', ' || v\_strada || ' ' || v\_numar;

EXCEPTION

WHEN NO\_ID\_FOUND THEN

RETURN 'Nu s-a gasit niciun pachet de vacanta pentru ID-ul specificat.';

WHEN TOO\_FEW\_STARS THEN

RETURN'S-a gasit pachetul, dar cazarea are 3 sau mai putine stele!';

WHEN OTHERS THEN

RETURN 'O eroare necunoscuta a aparut.';

END detalii\_pachet\_vacanta;

/

DECLARE

v\_mesaj VARCHAR2(200);

BEGIN

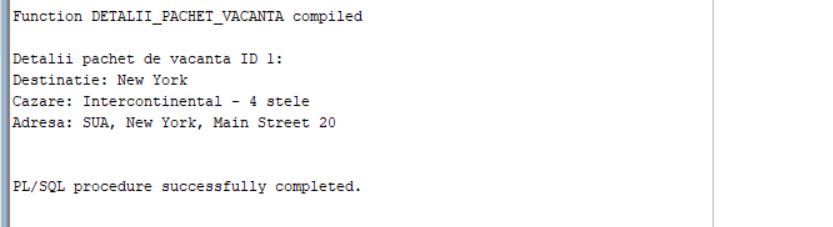
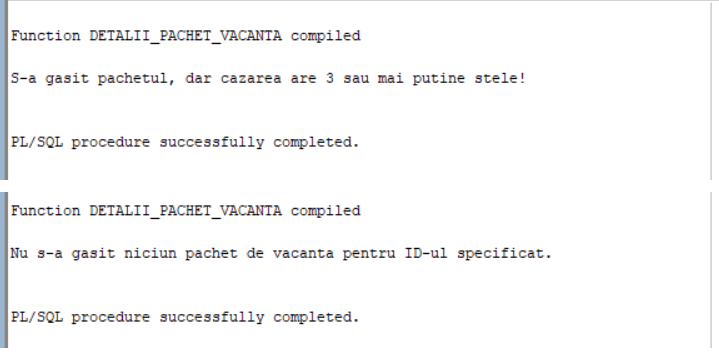
v\_mesaj := detalii\_pachet\_vacanta(1);

DBMS\_OUTPUT.PUT\_LINE(v\_mesaj);

END;

/





1. **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.**

Pentru un pachet de vacanta anume, sa se afiseze id-ul pachetului, destinatia, denumirea cazarii specifice pachetului de vacanta, numarul de stele cazare si adresa la care se afla cazarea respectiva (tara, oras, strada, numar). In plus, sa se afiseze id-ul, numele si prenumele angajatilor care au efectuat rezervari ce contin pachetul de vacanta respectiv. Daca nu exista pachet de vacanta cu id-ul specificat sa se ridice exceptia NO\_DATA\_FOUND, si daca exista mai multi angajati care au efectuat rezervari pentru pachetul de vacanta respectiv sa se ridice exceptia TOO\_MANY\_ROWS.

CREATE OR REPLACE PROCEDURE detalii\_pachet\_vacanta\_angajati (

id\_pachet\_param IN NUMBER

)

IS

TYPE EmpList IS TABLE OF NUMBER;

v\_id\_pachet PACHET\_VACANTA.id\_pachetvac%TYPE;

v\_destinatie PACHET\_VACANTA.destinatie%TYPE;

v\_denumire\_cazare CAZARE.denumire%TYPE;

v\_nr\_stele\_cazare CAZARE.nr\_stele%TYPE;

v\_tara ADRESA.tara%TYPE;

v\_oras ADRESA.oras%TYPE;

v\_strada ADRESA.strada%TYPE;

v\_numar ADRESA.numar%TYPE;

v\_displayed\_emp EmpList := EmpList(); -- Nested table pentru a contoriza angajatii

v\_numar\_inregistrari NUMBER := 0;

BEGIN

-- Initializare variabile pentru detalii pachet

v\_id\_pachet := NULL;

v\_destinatie := NULL;

v\_denumire\_cazare := NULL;

v\_nr\_stele\_cazare := NULL;

v\_tara := NULL;

v\_oras := NULL;

v\_strada := NULL;

v\_numar := NULL;

-- Utilizăm cele cinci tabele într-o singură comandă SQL

FOR rec IN (

SELECT pv.id\_pachetvac, pv.destinatie, c.denumire, c.nr\_stele, a.tara, a.oras, a.strada, a.numar, ang.id\_angajat, ang.nume, ang.prenume

FROM PACHET\_VACANTA pv

JOIN CAZARE c ON pv.id\_cazare = c.id\_cazare

JOIN ADRESA a ON c.id\_adresa = a.id\_adresa

JOIN REZERVARE r ON pv.id\_pachetvac = r.id\_pachetvac

JOIN ANGAJAT ang ON r.id\_angajat = ang.id\_angajat

WHERE pv.id\_pachetvac = id\_pachet\_param

ORDER BY ang.id\_angajat

) LOOP

-- Afișăm detalii despre pachet doar o singură dată

IF v\_id\_pachet IS NULL THEN

v\_id\_pachet := rec.id\_pachetvac;

v\_destinatie := rec.destinatie;

v\_denumire\_cazare := rec.denumire;

v\_nr\_stele\_cazare := rec.nr\_stele;

v\_tara := rec.tara;

v\_oras := rec.oras;

v\_strada := rec.strada;

v\_numar := rec.numar;

DBMS\_OUTPUT.PUT\_LINE('Detalii pachet de vacanta ID ' || v\_id\_pachet || ':');

DBMS\_OUTPUT.PUT\_LINE('Destinatie: ' || v\_destinatie);

DBMS\_OUTPUT.PUT\_LINE('Cazare: ' || v\_denumire\_cazare || ' - ' || v\_nr\_stele\_cazare || ' stele');

DBMS\_OUTPUT.PUT\_LINE('Adresa: ' || v\_tara || ', ' || v\_oras || ', ' || v\_strada || ' ' || v\_numar || CHR(10));

DBMS\_OUTPUT.PUT\_LINE('Angajatii care au efectuat rezervari cu pachetul de vacanta ID ' || v\_id\_pachet || ':');

END IF;

-- Afișăm informațiile despre angajat doar dacă nu a fost deja afișat

IF NOT v\_displayed\_emp.EXISTS(v\_displayed\_emp.LAST) OR v\_displayed\_emp(v\_displayed\_emp.LAST) <> rec.id\_angajat THEN

v\_numar\_inregistrari := v\_numar\_inregistrari + 1;

DBMS\_OUTPUT.PUT\_LINE('Angajat ID ' || rec.id\_angajat || ': ' || rec.nume || ' ' || rec.prenume);

v\_displayed\_emp.EXTEND;

v\_displayed\_emp(v\_displayed\_emp.LAST) := rec.id\_angajat;

END IF;

END LOOP;

-- Verificăm dacă s-au găsit înregistrări

IF v\_id\_pachet IS NULL THEN

RAISE NO\_DATA\_FOUND;

END IF;

-- Verificăm dacă s-au găsit prea multe înregistrări

IF v\_numar\_inregistrari > 1 THEN

RAISE TOO\_MANY\_ROWS;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au gasit detalii pentru pachetul de vacanta cu ID-ul specificat.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Mai mult de o rezervare gasita pentru pachetul de vacanta cu ID-ul specificat.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('O eroare necunoscută a apărut: ' || SQLERRM);

END detalii\_pachet\_vacanta\_angajati;

/

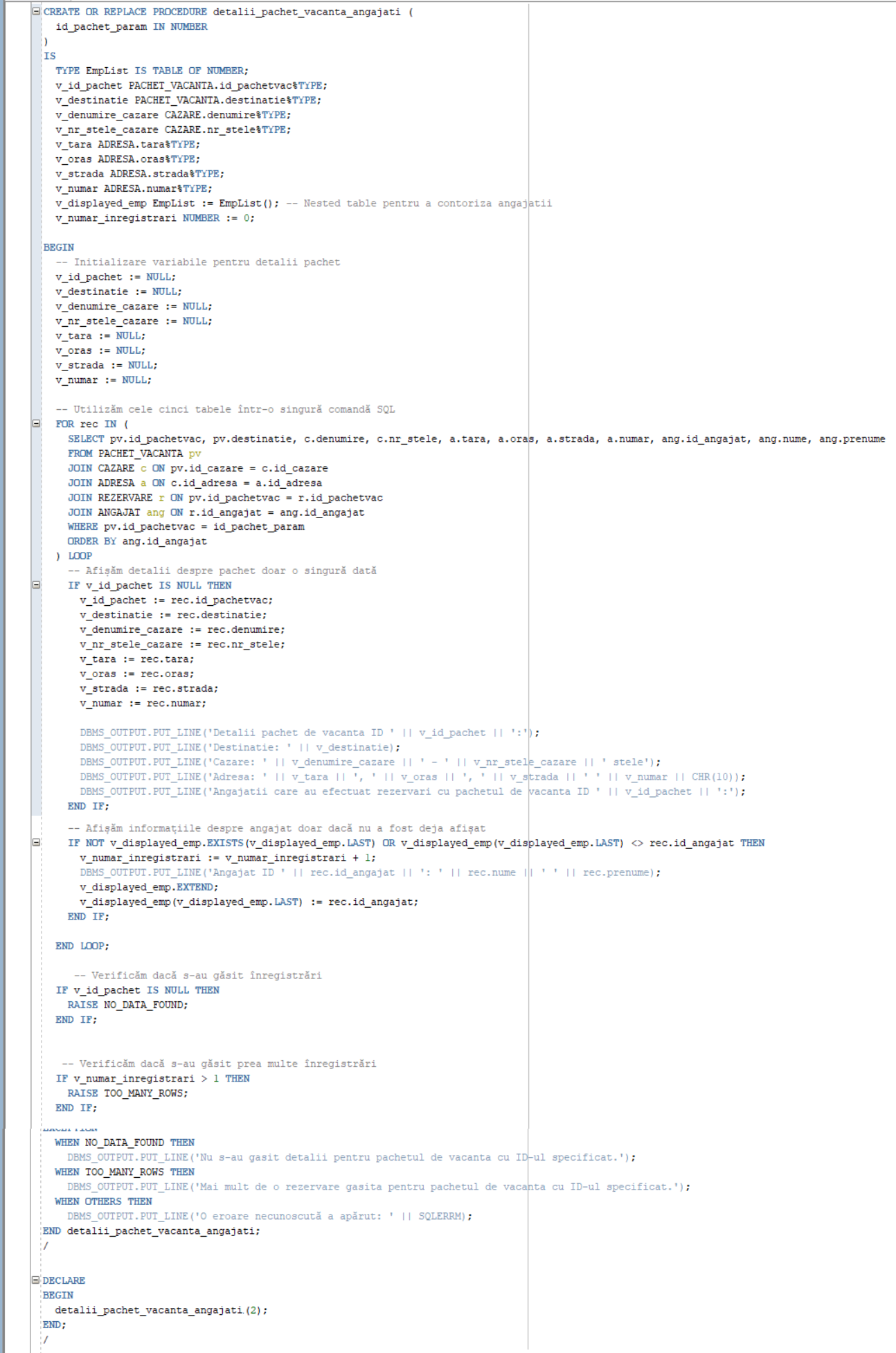
DECLARE

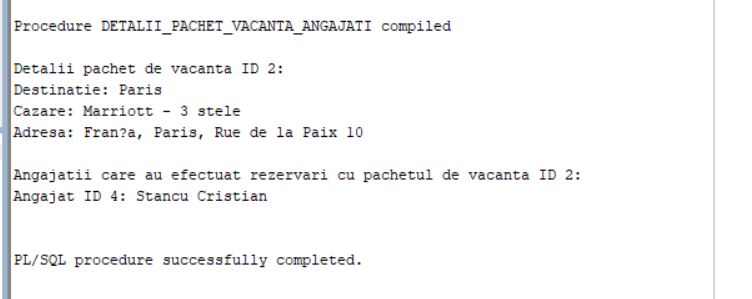
BEGIN

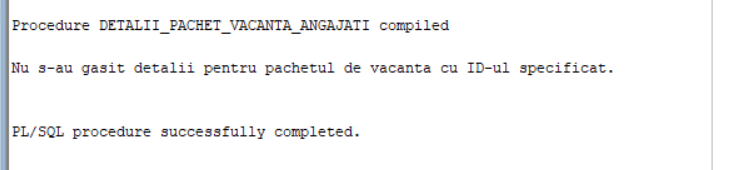
detalii\_pachet\_vacanta\_angajati(2);

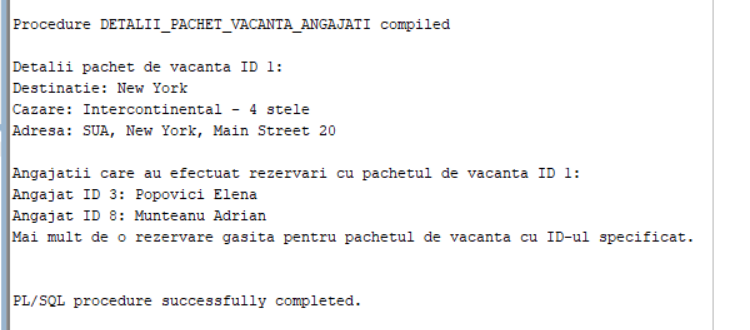
END;

/









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

Definiti un trigger LMD la nivel de comanda care blocheaza stergerea pachetului cu destinatia „New York”.

CREATE OR REPLACE TRIGGER trg\_prevent\_delete\_new\_york

BEFORE DELETE ON PACHET\_VACANTA

DECLARE

v\_count NUMBER;

BEGIN

-- Verificăm dacă există înregistrări cu destinația "New York" care urmează să fie șterse

SELECT COUNT(\*)

INTO v\_count

FROM PACHET\_VACANTA

WHERE destinatie = 'New York';

-- Dacă există astfel de înregistrări, anulăm operația DELETE și afișăm un mesaj de eroare

IF v\_count > 0 THEN

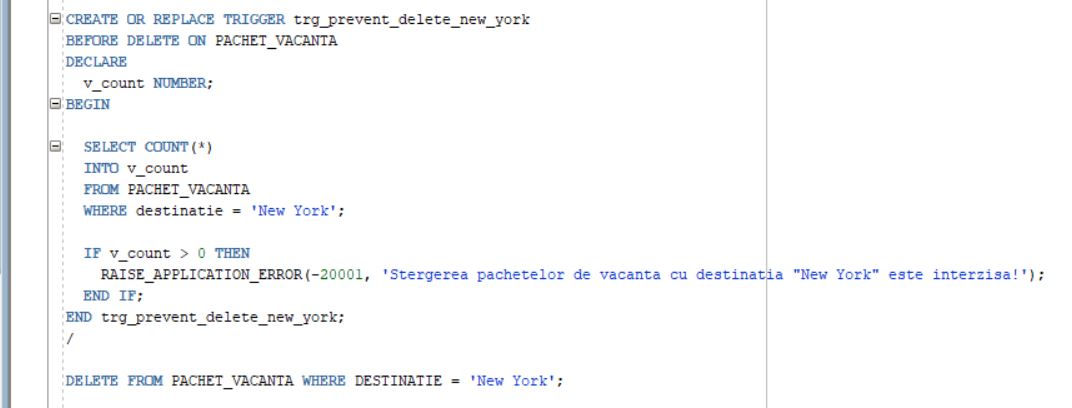
RAISE\_APPLICATION\_ERROR(-20001, 'Stergerea pachetelor de vacanta cu destinatia "New York" este interzisa!');

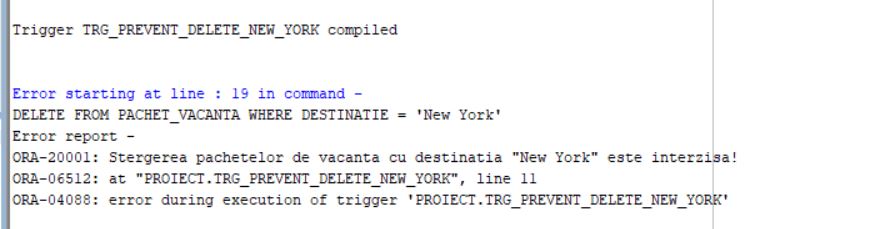
END IF;

END trg\_prevent\_delete\_new\_york;

/

DELETE FROM PACHET\_VACANTA WHERE DESTINATIE = 'New York';

7



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

Definiti un trigger de tip LMD la nivel de linie astfel incat atunci cand se schimba durata unui pachet de vacanta sa se modifice si pretul acestuia cu 10% pentru fiecare zi in plus/minus fata de durata initiala.

CREATE OR REPLACE TRIGGER TRG\_UPDATE\_PRET\_SEJUR

BEFORE UPDATE ON PACHET\_VACANTA

FOR EACH ROW

DECLARE

v\_durata\_veche NUMBER;

v\_durata\_noua NUMBER;

v\_pret\_vechi NUMBER;

BEGIN

v\_durata\_veche := :OLD.durata;

v\_durata\_noua := :NEW.durata;

v\_pret\_vechi := :OLD.pret\_sejur;

IF v\_durata\_noua <> v\_durata\_veche THEN

DECLARE

v\_diferenta\_durata NUMBER;

BEGIN

v\_diferenta\_durata := v\_durata\_noua - v\_durata\_veche;

:NEW.pret\_sejur := :NEW.pret\_sejur + (v\_diferenta\_durata \* (v\_pret\_vechi \* 0.1));

END;

END IF;

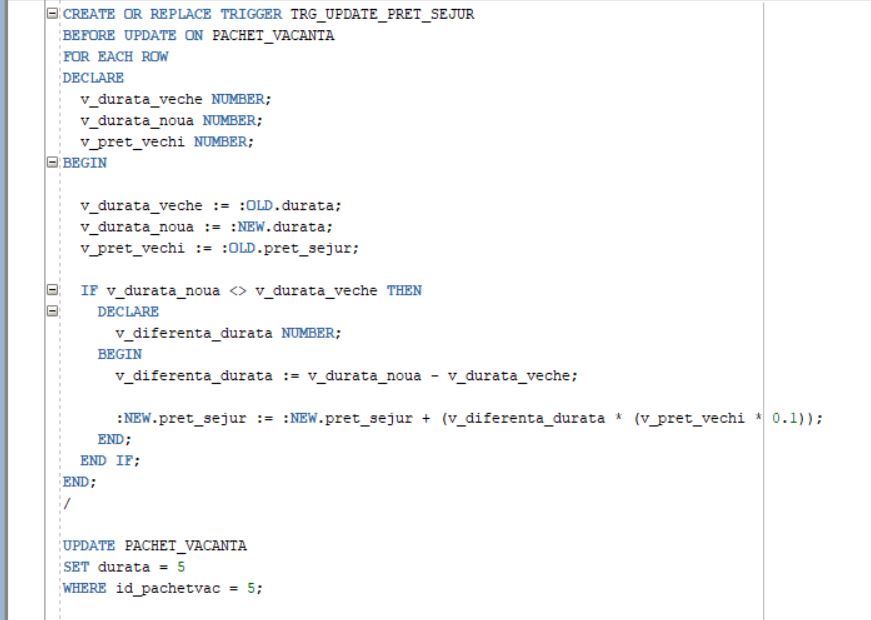
END;

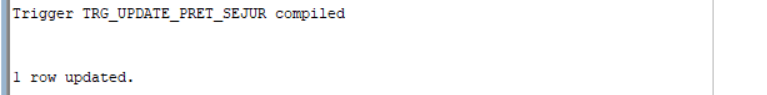
/

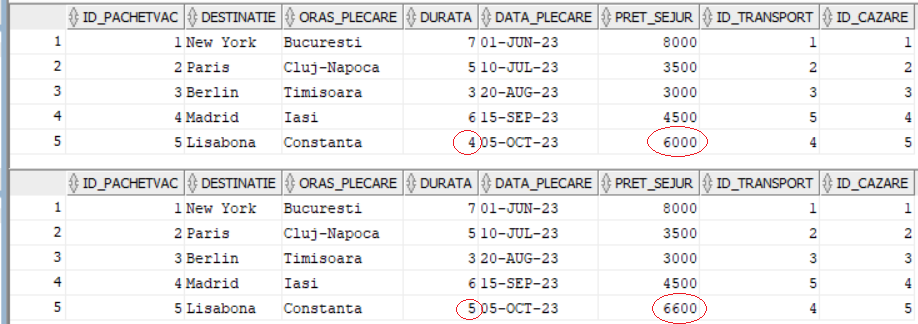
UPDATE PACHET\_VACANTA

SET durata = 5

WHERE id\_pachetvac = 5;







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

Definiti un trigger de tip LDD care sa nu permita stergerea tabelului „REZERVARE” din baza de date.

CREATE OR REPLACE TRIGGER trg\_prevent\_drop\_rezervari

BEFORE DROP ON SCHEMA

DECLARE

v\_object\_name VARCHAR2(30);

BEGIN

v\_object\_name := ora\_dict\_obj\_name;

IF ora\_dict\_obj\_type = 'TABLE' AND v\_object\_name = 'REZERVARE' THEN

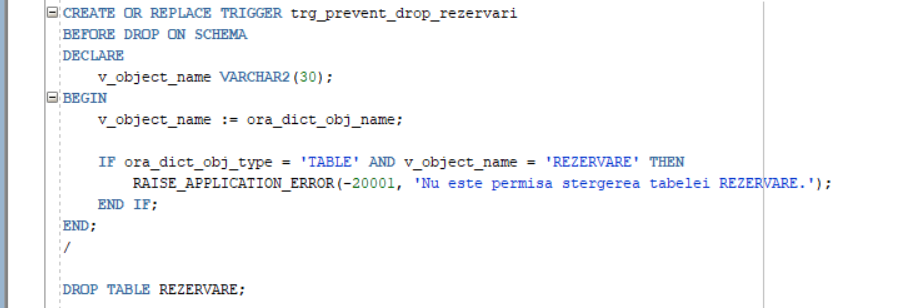
RAISE\_APPLICATION\_ERROR(-20001, 'Nu este permisa stergerea tabelei REZERVARE.');

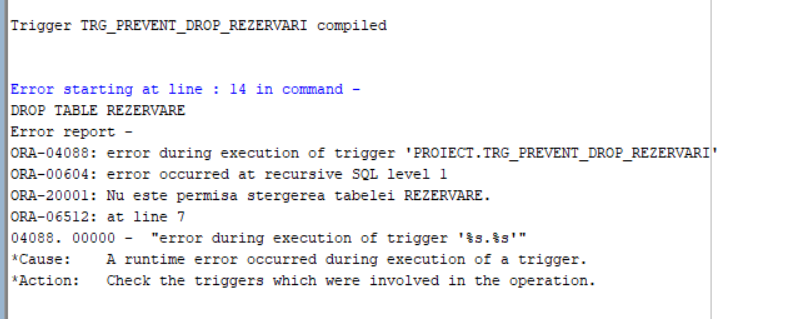
END IF;

END;

/

DROP TABLE REZERVARE;





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

CREATE OR REPLACE PACKAGE PACHET\_TURISM AS

PROCEDURE afisare\_informatii\_rezervare(p\_id\_rezervare IN NUMBER);

PROCEDURE detalii\_rezervari(p\_data\_inceput IN DATE, p\_data\_sfarsit IN DATE);

FUNCTION detalii\_pachet\_vacanta(id\_pachet\_param IN NUMBER) RETURN VARCHAR2;

PROCEDURE detalii\_pachet\_vacanta\_angajati(id\_pachet\_param IN NUMBER);

END PACHET\_TURISM;

/

CREATE OR REPLACE PACKAGE BODY PACHET\_TURISM AS

-- PROBLEMA 6

PROCEDURE afisare\_informatii\_rezervare(p\_id\_rezervare IN NUMBER) AS

v\_data\_rez DATE;

v\_nr\_pers NUMBER;

-- Folosirea tablourilor imbricate (NESTED TABLES) pentru detaliile pachetului de vacanta

TYPE DetaliiPachetTable IS TABLE OF Pachet\_vacanta%ROWTYPE;

v\_detalii\_pachet DetaliiPachetTable;

-- Folosirea tablourilor indexate (INDEX BY TABLES) pentru detaliile rezervarii

TYPE DetaliiRezervareIndexByTable IS TABLE OF Rezervare%ROWTYPE INDEX BY PLS\_INTEGER;

v\_detalii\_rezervare DetaliiRezervareIndexByTable;

-- Folosirea vectorilor (VARRAYS) pentru detaliile clientului

TYPE DetaliiClientVarray IS VARRAY(5) OF Client%ROWTYPE;

v\_detalii\_client DetaliiClientVarray := DetaliiClientVarray();

BEGIN

-- Obtine informatii despre rezervare

SELECT data\_rez, nr\_pers

INTO v\_data\_rez, v\_nr\_pers

FROM Rezervare

WHERE id\_rezervare = p\_id\_rezervare;

-- Obtine informatii despre pachetul de vacanta din rezervare folosind tablourile imbricate (NESTED TABLES)

SELECT p.\*

BULK COLLECT INTO v\_detalii\_pachet

FROM Pachet\_vacanta p

WHERE p.id\_pachetvac = (SELECT id\_pachetvac FROM Rezervare WHERE id\_rezervare = p\_id\_rezervare);

-- Obtine informatii despre rezervare folosind tablourile indexate (INDEX BY TABLES)

SELECT r.\*

INTO v\_detalii\_rezervare(1)

FROM Rezervare r

WHERE r.id\_rezervare = p\_id\_rezervare;

-- Obtine informatii despre client folosind vectorii (VARRAYS)

SELECT c.\*

BULK COLLECT INTO v\_detalii\_client

FROM Client c

WHERE c.id\_client = (SELECT id\_client FROM Rezervare WHERE id\_rezervare = p\_id\_rezervare);

-- Afiseaza informatii despre rezervare folosind tablourile indexate (INDEX BY TABLES)

DBMS\_OUTPUT.PUT\_LINE('Detalii rezervare:');

DBMS\_OUTPUT.PUT\_LINE(' Numar persoane: ' || v\_detalii\_rezervare(1).nr\_pers);

DBMS\_OUTPUT.PUT\_LINE(' Data rezervare: ' || TO\_CHAR(v\_detalii\_rezervare(1).data\_rez, 'YYYY-MM-DD'));

-- Afiseaza informatii despre pachet folosind tablourile imbricate (NESTED TABLES)

FOR i IN 1..v\_detalii\_pachet.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Detalii pachet:');

DBMS\_OUTPUT.PUT\_LINE(' Destinatie: ' || v\_detalii\_pachet(i).destinatie);

DBMS\_OUTPUT.PUT\_LINE(' Oras plecare: ' || v\_detalii\_pachet(i).oras\_plecare);

DBMS\_OUTPUT.PUT\_LINE(' Durata: ' || v\_detalii\_pachet(i).durata || ' zile');

DBMS\_OUTPUT.PUT\_LINE(' Data plecare: ' || TO\_CHAR(v\_detalii\_pachet(i).data\_plecare, 'YYYY-MM-DD'));

DBMS\_OUTPUT.PUT\_LINE(' Pret sejur: ' || v\_detalii\_pachet(i).pret\_sejur);

END LOOP;

-- Afiseaza informatii despre client folosind vectorii (VARRAYS)

FOR i IN 1..v\_detalii\_client.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Detalii client care a efectuat rezervarea:');

DBMS\_OUTPUT.PUT\_LINE(' Nume: ' || v\_detalii\_client(i).nume);

DBMS\_OUTPUT.PUT\_LINE(' Prenume: ' || v\_detalii\_client(i).prenume);

DBMS\_OUTPUT.PUT\_LINE(' Sex: ' || v\_detalii\_client(i).sex);

DBMS\_OUTPUT.PUT\_LINE(' Numar de telefon: ' || v\_detalii\_client(i).nr\_tel);

DBMS\_OUTPUT.PUT\_LINE(' Email: ' || v\_detalii\_client(i).email);

DBMS\_OUTPUT.PUT\_LINE(' Data nastere: ' || TO\_CHAR(v\_detalii\_client(i).data\_n, 'YYYY-MM-DD'));

END LOOP;

END;

-- PROBLEMA 7

PROCEDURE detalii\_rezervari(p\_data\_inceput IN DATE, p\_data\_sfarsit IN DATE) AS

-- Cursor clasic pentru a obține rezervările în intervalul de date specificat

CURSOR cur\_rezervari\_clasic IS

SELECT id\_rezervare, id\_client, data\_rez

FROM rezervare

WHERE data\_rez BETWEEN p\_data\_inceput AND p\_data\_sfarsit;

-- Cursor parametrizat pentru a obține detalii suplimentare despre fiecare rezervare

CURSOR cur\_detalii\_rezervare (p\_id\_rezervare NUMBER) IS

SELECT r.id\_rezervare, c.nume, c.prenume, p.destinatie, p.pret\_sejur, r.data\_rez

FROM rezervare r

JOIN client c ON r.id\_client = c.id\_client

JOIN pachet\_vacanta p ON r.id\_pachetvac = p.id\_pachetvac

WHERE r.id\_rezervare = p\_id\_rezervare;

-- Variabile pentru stocarea rezultatelor

v\_id\_rezervare NUMBER;

v\_nume\_client VARCHAR2(30);

v\_prenume\_client VARCHAR2(30);

v\_destinatie VARCHAR2(20);

v\_pret\_sejur NUMBER;

v\_data\_rez DATE;

BEGIN

-- Deschidem cursorul clasic și iterăm prin rezervări

FOR rez IN cur\_rezervari\_clasic LOOP

v\_id\_rezervare := rez.id\_rezervare;

-- Deschidem cursorul parametrizat și obținem detalii despre rezervare

FOR detalii IN cur\_detalii\_rezervare(v\_id\_rezervare) LOOP

v\_nume\_client := detalii.nume;

v\_prenume\_client := detalii.prenume;

v\_destinatie := detalii.destinatie;

v\_pret\_sejur := detalii.pret\_sejur;

v\_data\_rez := detalii.data\_rez;

-- Afișăm sau stocăm rezultatele cum este necesar

DBMS\_OUTPUT.PUT\_LINE('Rezervare ID: ' || v\_id\_rezervare);

DBMS\_OUTPUT.PUT\_LINE('Data rezervare: ' || v\_data\_rez);

DBMS\_OUTPUT.PUT\_LINE('Nume client: ' || v\_nume\_client || ' ' || v\_prenume\_client);

DBMS\_OUTPUT.PUT\_LINE('Destinatie: ' || v\_destinatie);

DBMS\_OUTPUT.PUT\_LINE('Pret sejur: ' || v\_pret\_sejur);

DBMS\_OUTPUT.PUT\_LINE('-------------------');

END LOOP;

END LOOP;

END detalii\_rezervari;

-- PROBLEMA 8

FUNCTION detalii\_pachet\_vacanta (

id\_pachet\_param IN NUMBER

) RETURN VARCHAR2

IS

v\_cursor SYS\_REFCURSOR;

v\_id\_pachet NUMBER;

v\_destinatie VARCHAR2(20);

v\_denumire\_cazare VARCHAR2(20);

v\_nr\_stele\_cazare NUMBER;

v\_tara VARCHAR2(20);

v\_oras VARCHAR2(20);

v\_strada VARCHAR2(20);

v\_numar NUMBER;

NO\_ID\_FOUND EXCEPTION;

TOO\_FEW\_STARS EXCEPTION;

BEGIN

OPEN v\_cursor FOR

SELECT pv.id\_pachetvac, pv.destinatie, c.denumire, c.nr\_stele, a.tara, a.oras, a.strada, a.numar

FROM PACHET\_VACANTA pv

JOIN CAZARE c ON pv.id\_cazare = c.id\_cazare

JOIN ADRESA a ON c.id\_adresa = a.id\_adresa

WHERE pv.id\_pachetvac = id\_pachet\_param;

FETCH v\_cursor INTO v\_id\_pachet, v\_destinatie, v\_denumire\_cazare, v\_nr\_stele\_cazare, v\_tara, v\_oras, v\_strada, v\_numar;

IF v\_cursor%NOTFOUND THEN

CLOSE v\_cursor;

RAISE NO\_ID\_FOUND;

END IF;

IF v\_nr\_stele\_cazare <= 3 THEN

CLOSE v\_cursor;

RAISE TOO\_FEW\_STARS;

END IF;

CLOSE v\_cursor;

RETURN 'Detalii pachet de vacanta ID ' || v\_id\_pachet || ':' || CHR(10) ||

'Destinatie: ' || v\_destinatie || CHR(10) ||

'Cazare: ' || v\_denumire\_cazare || ' - ' || v\_nr\_stele\_cazare || ' stele' || CHR(10) ||

'Adresa: ' || v\_tara || ', ' || v\_oras || ', ' || v\_strada || ' ' || v\_numar;

EXCEPTION

WHEN NO\_ID\_FOUND THEN

RETURN 'Nu s-a gasit niciun pachet de vacanta pentru ID-ul specificat.';

WHEN TOO\_FEW\_STARS THEN

RETURN'S-a gasit pachetul, dar cazarea are 3 sau mai putine stele!';

WHEN OTHERS THEN

RETURN 'O eroare necunoscuta a aparut.';

END detalii\_pachet\_vacanta;

-- PROBLEMA 9

PROCEDURE detalii\_pachet\_vacanta\_angajati(id\_pachet\_param IN NUMBER) AS

TYPE EmpList IS TABLE OF NUMBER;

v\_id\_pachet PACHET\_VACANTA.id\_pachetvac%TYPE;

v\_destinatie PACHET\_VACANTA.destinatie%TYPE;

v\_denumire\_cazare CAZARE.denumire%TYPE;

v\_nr\_stele\_cazare CAZARE.nr\_stele%TYPE;

v\_tara ADRESA.tara%TYPE;

v\_oras ADRESA.oras%TYPE;

v\_strada ADRESA.strada%TYPE;

v\_numar ADRESA.numar%TYPE;

v\_displayed\_emp EmpList := EmpList(); -- Nested table pentru a contoriza angajatii

BEGIN

-- Initializare variabile pentru detalii pachet

v\_id\_pachet := NULL;

v\_destinatie := NULL;

v\_denumire\_cazare := NULL;

v\_nr\_stele\_cazare := NULL;

v\_tara := NULL;

v\_oras := NULL;

v\_strada := NULL;

v\_numar := NULL;

-- Utilizăm cele cinci tabele într-o singură comandă SQL

FOR rec IN (

SELECT pv.id\_pachetvac, pv.destinatie, c.denumire, c.nr\_stele, a.tara, a.oras, a.strada, a.numar, ang.id\_angajat, ang.nume, ang.prenume

FROM PACHET\_VACANTA pv

JOIN CAZARE c ON pv.id\_cazare = c.id\_cazare

JOIN ADRESA a ON c.id\_adresa = a.id\_adresa

JOIN REZERVARE r ON pv.id\_pachetvac = r.id\_pachetvac

JOIN ANGAJAT ang ON r.id\_angajat = ang.id\_angajat

WHERE pv.id\_pachetvac = id\_pachet\_param

ORDER BY ang.id\_angajat

) LOOP

-- Afișăm detalii despre pachet doar o singură dată

IF v\_id\_pachet IS NULL THEN

v\_id\_pachet := rec.id\_pachetvac;

v\_destinatie := rec.destinatie;

v\_denumire\_cazare := rec.denumire;

v\_nr\_stele\_cazare := rec.nr\_stele;

v\_tara := rec.tara;

v\_oras := rec.oras;

v\_strada := rec.strada;

v\_numar := rec.numar;

DBMS\_OUTPUT.PUT\_LINE('Detalii pachet de vacanta ID ' || v\_id\_pachet || ':');

DBMS\_OUTPUT.PUT\_LINE('Destinatie: ' || v\_destinatie);

DBMS\_OUTPUT.PUT\_LINE('Cazare: ' || v\_denumire\_cazare || ' - ' || v\_nr\_stele\_cazare || ' stele');

DBMS\_OUTPUT.PUT\_LINE('Adresa: ' || v\_tara || ', ' || v\_oras || ', ' || v\_strada || ' ' || v\_numar || CHR(10));

DBMS\_OUTPUT.PUT\_LINE('Angajatii care au efectuat rezervari cu pachetul de vacanta ID ' || v\_id\_pachet || ':');

END IF;

-- Afișăm informațiile despre angajat doar dacă nu a fost deja afișat

IF NOT v\_displayed\_emp.EXISTS(v\_displayed\_emp.LAST) OR v\_displayed\_emp(v\_displayed\_emp.LAST) <> rec.id\_angajat THEN

DBMS\_OUTPUT.PUT\_LINE('Angajat ID ' || rec.id\_angajat || ': ' || rec.nume || ' ' || rec.prenume);

v\_displayed\_emp.EXTEND; -- Extend the nested table

v\_displayed\_emp(v\_displayed\_emp.LAST) := rec.id\_angajat;

END IF;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au găsit detalii pentru pachetul de vacanță cu ID-ul specificat.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Prea multe înregistrări găsite pentru pachetul de vacanță cu ID-ul specificat.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('O eroare necunoscută a apărut: ' || SQLERRM);

END detalii\_pachet\_vacanta\_angajati;

END PACHET\_TURISM;

/

