

PROIECT SGBD

**Tămaș Iulia
grupa 233**

Cerinta 1.

Prezentați pe scurt baza de date (utilitatea ei)

Proiectul realizat se referă la proiectarea unei baze de date ce stochează informații despre funcionalitatea mai multor depozite de vehicule. Aceste depozite au locații unice care se găsesc în mai multe provincii și mai multe țări de pe 6 continente (Europa, Asia, Africa, America de sud și de nord și Australia).

Fiecare depozit are un unic proprietar care este considerat o persoana fizica cu atribute caracteristice. Un proprietar poate detine mai multe depozite.

In cadrul unui depozit se află mai mulți angajați distribuiți pe cele 7 locuri de munca existente și anume: mecanic auto, inspector tehnic, operator de stocare, sofer de livrare, tehnician de vopsire, paznic și contabil. Toate aceste locuri de munca au setat un minim și maxim salarial. Angajatii pot lucra la mai multe depozite pe joburi diferite. In cadrul bazei de date se află mai multe detalii despre angajarea unei persoane cum ar fi data angajării și salariul.

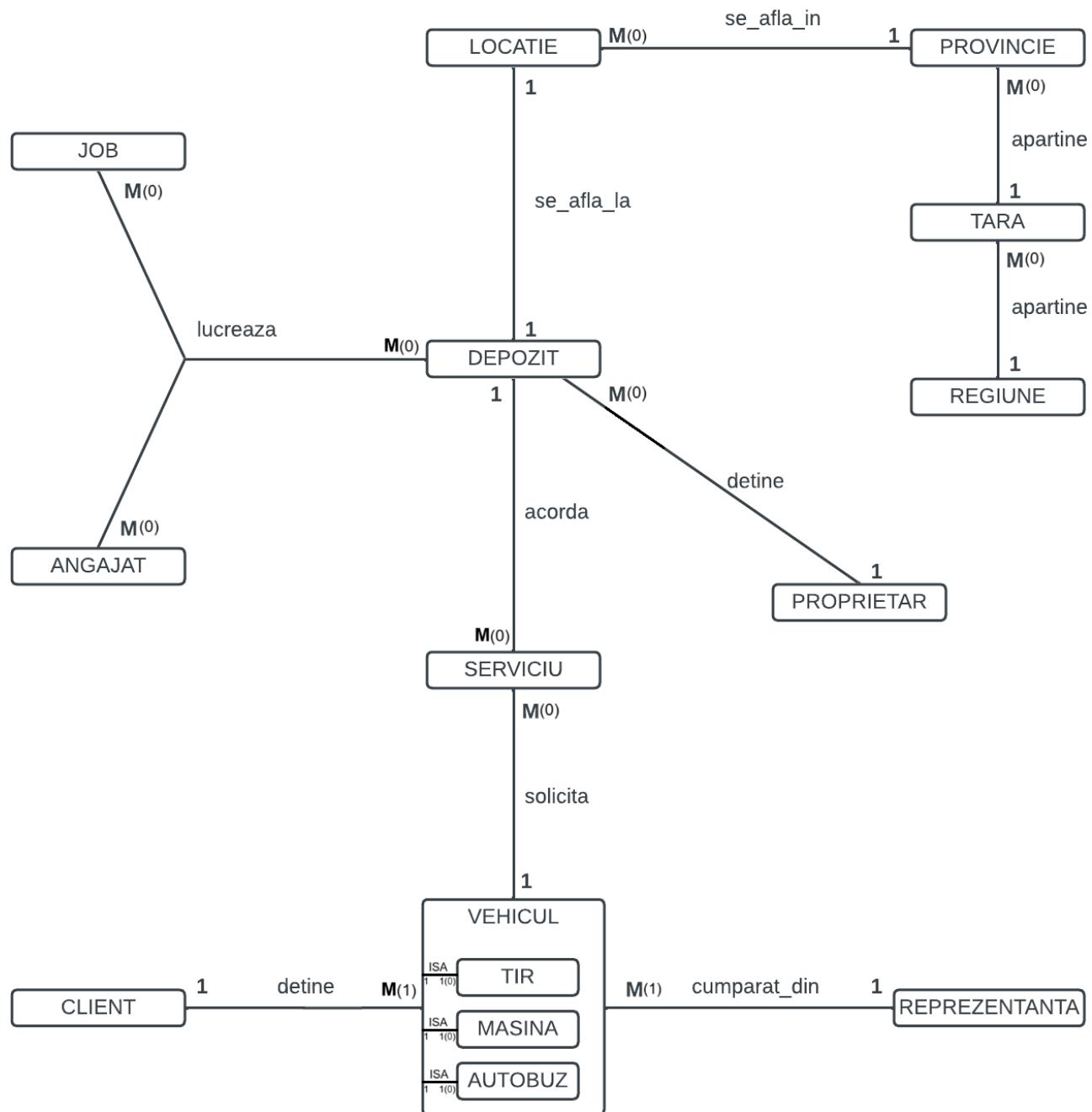
Un depozit poate pune la dispozitie mai multe servicii pentru vehicule în funcție de necesitate. Câteva exemple de servicii disponibile pentru vehicule sunt: vopsire, curătare chimică, schimb ulei. Cand un vehicul necesita unul din aceste servicii este trecut în baza de date cu informațiile corespunzătoare precum id-ul depozitului care îi va oferi serviciul, id-ul vehiculului, denumirea explicită a serviciului de care va beneficia vehiculului, data de introducere în depozit și data cand lucrarea este finalizată.

Vehiculele sunt grupate în 3 categorii și anume mașina, autobuz și tir. Vehiculele au anumite trăsături generale urmând ca fiecare tipologie să aibă atribuții specifice. Fiecare vehicul are un unic client/detinator care este considerat o persoana fizica cu caracteristici proprii. Un client poate detine mai multe vehicule din mai multe categorii. Fiecare vehicul este cumpărat dintr-o reprezentantă auto, nu neapărat determinată de marca vehiculului. O reprezentantă are atribuții caracteristice precum id-ul reprezentantei, numele complet și contact. Din reprezentanta se pot cumpăra mai multe vehicule însă un vehicul poate fi cumpărat dintr-o singură reprezentantă.

Cerinta 2.

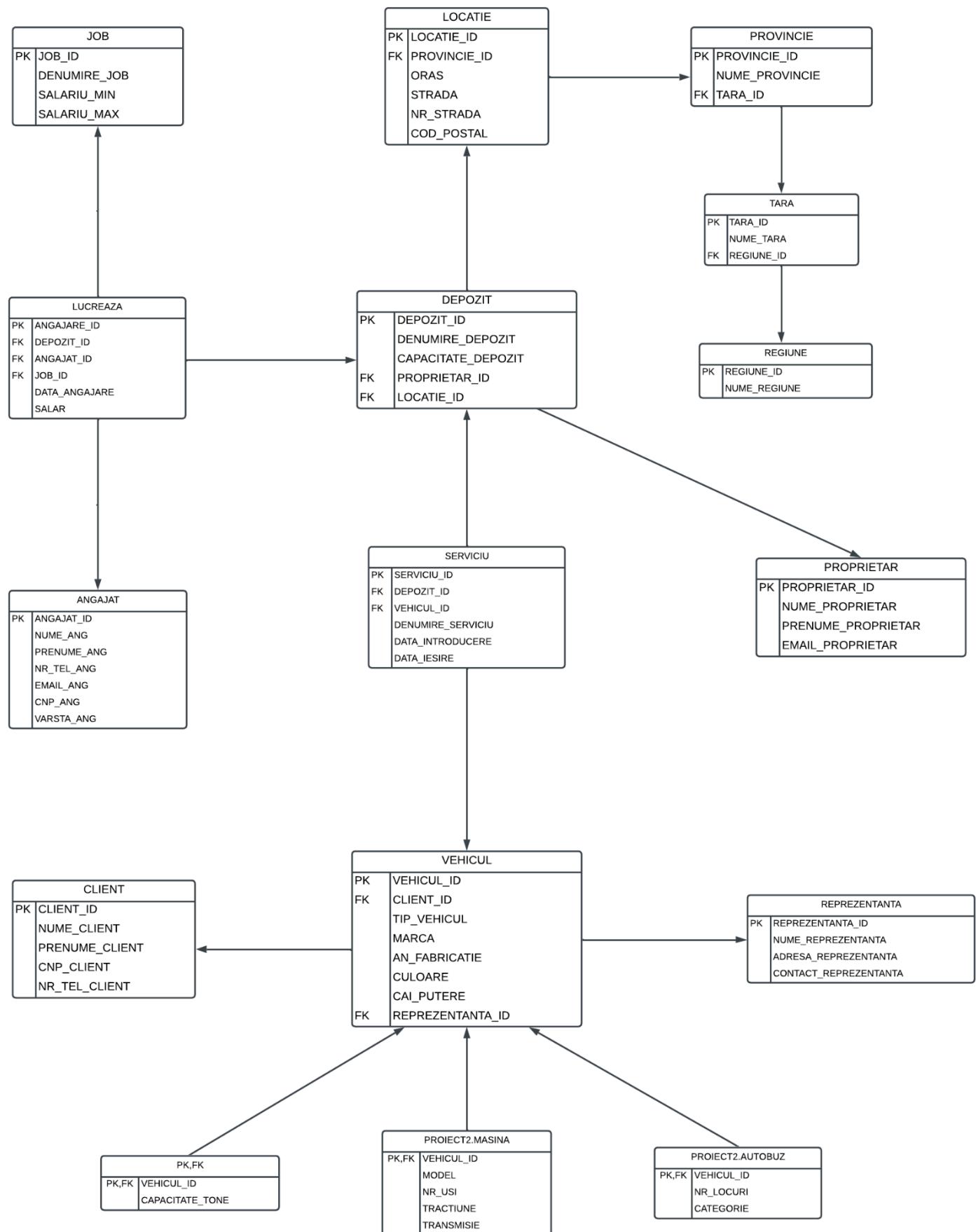
Realizați diagrama entitate-relație (ERD)

Diagrama entitate-relație corespunzătoare bazei de date create este aceasta:



Cerinta 3.

Pornind de la diagrama entitate-relație realizăți diagrama conceptuală a modelului propus



Cerinta 4.

Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare

```
CREATE TABLE regiune (
    regiune_id number constraint regiune_id_nn not null,
    nume_regiune varchar2(15)
);
```

```
CREATE UNIQUE INDEX regiune_id_pk
ON regiune (regiune_id);
```

```
ALTER TABLE regiune
ADD (CONSTRAINT regiune_id_pk PRIMARY KEY (regiune_id)) ;
```

```
CREATE TABLE tara (
    tara_id char(2) constraint tara_id_nn not null,
    nume_tara varchar2(15),
    regiune_id number,
    constraint tara_c_id_pk PRIMARY KEY (tara_id)
);
```

```
ALTER TABLE tara
ADD (CONSTRAINT tara_reg_fk FOREIGN KEY(regiune_id) REFERENCES regiune(regiune_id));
```

```
CREATE TABLE provincie (
    provincie_id char(2) constraint provincie_id_nn not null,
    nume_provincie varchar2(15),
    tara_id char(2),
    constraint provincie_id_pk PRIMARY KEY (provincie_id)
);
```

```
ALTER TABLE provincie
ADD (CONSTRAINT provincie_tara_fk FOREIGN KEY(tara_id) REFERENCES tara(tara_id));
```

```
CREATE TABLE locatie (
    locatie_id number(4),
    provincie_id char(2),
    oras varchar2(30) CONSTRAINT loc_oras_nn NOT NULL,
    strada varchar2(30),
    nr_strada number,
    cod_postal varchar2(10)
);
```

```
CREATE UNIQUE INDEX loc_id_pk
ON locatie (locatie_id) ;
```

```
alter table locatie
add (constraint loc_id_pk primary key (locatie_id),
constraint loc_prov_id_fk foreign key (provincie_id) references provincie(provincie_id),
constraint cod_pos_unq unique(cod_postal));
```

```
CREATE TABLE proprietar (
    proprietar_id number(3) constraint proprietar_id_nn not null,
    nume_proprietar varchar2(20),
    prenume_proprietar varchar2(20),
    email_proprietar varchar2(30)
);
```

```
alter table proprietar  
add (constraint propri_id_pk primary key (proprietar_id));
```

```
CREATE TABLE client (  
client_id number constraint client_id_nn not null,  
nume_client varchar2(20),  
prenume_client varchar2(20),  
cnp_client number(13) constraint cnp_client_unq unique,  
nr_tel_client varchar2(10)  
);
```

```
alter table client  
add (constraint client_id_pk primary key (client_id));
```

```
CREATE TABLE reprezentanta (  
reprezentanta_id number constraint reprezentanta_id_nn not null,  
nume_reprezentanta varchar2(30),  
adresa_reprezentanta varchar2(30),  
contact_reprezentanta varchar2(10)  
);
```

```
alter table reprezentanta  
add (constraint reprezentanta_id_pk primary key (reprezentanta_id));
```

```
CREATE TABLE angajat (  
angajat_id number constraint angajat_id_nn not null,  
nume_ang varchar2(20),  
prenume_ang varchar2(20),  
nr_tel_ang varchar2(10),  
email_ang varchar2(30) constraint email_ang_unq unique,  
cnp_ang varchar2(30) constraint cnp_ang_unq unique,  
varsta_ang number  
);
```

```
CREATE UNIQUE INDEX angajat_id_pk  
ON angajat (angajat_id) ;
```

```
alter table angajat  
add (constraint angajat_id_pk primary key (angajat_id));
```

```
create table job (  
job_id varchar2(10)constraint job_id_nn not null,  
denumire_job varchar2(30) constraint den_job_nn not null,  
salariu_min number,  
salariu_max number  
);
```

```
CREATE UNIQUE INDEX job_id_pk  
ON job (job_id) ;
```

```
ALTER TABLE job  
ADD (CONSTRAINT job_id_pk PRIMARY KEY(job_id));
```

```
create table vehicul (  
vehicul_id number constraint vehicul_id_nn not null,  
client_id number,  
tip_vehicul varchar2(20) constraint tip_veh_nn not null,  
marca varchar2(20),  
an_fabricatie number(4),  
culoare varchar2(20),
```

```
cai_putere number,  
reprezentanta_id number  
);
```

```
alter table vehicul  
add (constraint vehicul_id_pk primary key (vehicul_id),  
constraint veh_client_id_fk foreign key(client_id) references client(client_id),  
constraint vehic_tip_pk unique (vehicul_id, tip_vehicul),  
constraint veh_tip_ck check (lower(tip_vehicul) in ('masina', 'autobuz', 'tir')),  
constraint veh_reprezentanta_id_fk foreign key(reprezentanta_id) references reprezentanta(reprezentanta_id)  
);
```

```
create table masina (  
vehicul_id number constraint veh_mas_id_pk primary key,  
--tip_vehicul varchar2(20) default 'masina' constraint tip_mas_ck check (lower(tip_vehicul) = 'masina'),  
model varchar2(20),  
nr_usi number(1),  
tractiune varchar2(15),  
transmisie varchar2(15)  
);
```

```
alter table masina  
add (constraint veh_mas_fk foreign key (vehicul_id)  
references vehicul(vehicul_id));
```

```
create table autobuz (  
vehicul_id number constraint veh_bus_id_pk primary key,  
--tip_vehicul varchar2(20) default 'autobuz' constraint tip_bus_ck check (lower(tip_vehicul) = 'autobuz'),  
nr_locuri number(3),  
categorie varchar2(15)  
);
```

```
alter table autobuz  
add (constraint veh_bus_fk foreign key (vehicul_id)  
references vehicul(vehicul_id));
```

```
create table tir (  
vehicul_id number constraint veh_tir_id_pk primary key,  
--tip_vehicul varchar2(20) default 'tir' constraint tip_tir_ck check (lower(tip_vehicul) = 'tir'),  
capacitate_tone number(3)  
);
```

```
alter table tir  
add (constraint veh_tir_fk foreign key (vehicul_id)  
references vehicul(vehicul_id));
```

```
create table depozit(  
depozit_id number(4),  
denumire_depozit varchar2(30) constraint depo_den_nn not null,  
capacitate_depozit number,  
proprietar_id number(3),  
locatie_id number(4)  
);
```

```
CREATE UNIQUE INDEX depo_id_pk  
ON depozit (depozit_id) ;
```

```
alter table depozit  
add (constraint depo_id_pk primary key(depozit_id),  
constraint depo_prop_fk foreign key(proprietar_id) references proprietar(proprietar_id),  
constraint depo_loc_fk foreign key(locatie_id) references locatie(locatie_id),  
constraint depo_loc_unq unique(locatie_id);
```

```
-----  
create table serviciu (  
serviciu_id number(4),  
depozit_id number(4),  
vehicul_id number,  
denumire_serviciu varchar2(30) constraint serv_den_nn not null,  
data_introducere date,  
data_iesire date  
);
```

```
CREATE UNIQUE INDEX serv_id_pk  
ON serviciu (serviciu_id);
```

```
alter table serviciu  
add (constraint serv_id_pk primary key(serviciu_id),  
constraint serv_dep_fk foreign key(depozit_id) references depozit(depozit_id),  
constraint serv_veh_fk foreign key(vehicul_id) references vehicul(vehicul_id),  
constraint serv_interval check (data_introducere <= data_iesire) );  
-----
```

```
create table lucreaza (  
angajare_id number constraint angj_id_nn not null,  
depozit_id number(4) constraint angi_depo_id_nn not null,  
angajat_id number constraint angi_ang_id_nn not null,  
job_id varchar2(10) constraint angi_job_id_nn not null,  
data_angajare date constraint angi_data_nn not null,  
salar number  
);
```

```
CREATE UNIQUE INDEX angi_id_pk  
ON lucreaza(angajare_id);
```

```
alter table lucreaza  
add (constraint angi_id_pk primary key (angajare_id),  
constraint angi_depo_id_fk foreign key (depozit_id)  
references depozit(depozit_id),  
constraint angi_ang_id_fk foreign key (angajat_id)  
references angajat(angajat_id),  
constraint angi_job_id_fk foreign key (job_id)  
references job(job_id)  
);
```

```
commit;
```

Urmeaza printscreen-uri doveditoare cu faptul ca tabelele au fost create cu succes.

The screenshot shows the Oracle SQL Developer interface with the 'Worksheet' tab selected. The code area contains the SQL script for creating the 'regiune' and 'tara' tables, including their primary keys and foreign key constraints. The 'Script Output' pane below shows the successful execution of the script, indicating that both tables were created and altered successfully.

```
CREATE TABLE regiune (
    regiune_id number constraint regiune_id_nn not null,
    nume_regiune varchar2(15)
);

CREATE UNIQUE INDEX regiune_id_pk
ON regiune (regiune_id);

ALTER TABLE regiune
ADD (CONSTRAINT regiune_id_pk PRIMARY KEY (regiune_id)) ;

-----


CREATE TABLE tara (
    tara_id char(2) constraint tara_id_nn not null,
    nume_tara varchar2(15),
    regiune_id number,
    constraint tara_c_id_pk PRIMARY KEY (tara_id)
);

ALTER TABLE tara
ADD (CONSTRAINT tara_reg_fk FOREIGN KEY(regiune_id) REFERENCES regiune(regiune_id));
```

Script Output:

```
Table REGIUNE created.

INDEX REGIUNE_ID_PK created.

Table REGIUNE altered.

Table TARA created.

Table TARA altered.
```

The screenshot shows the Oracle SQL Developer interface with the 'Worksheet' tab selected. The code area contains the SQL script for creating the 'provincie' and 'locatie' tables, along with their indexes and foreign key constraints. The 'Script Output' pane below shows the successful execution of the script, indicating that both tables were created and altered successfully.

```
CREATE TABLE provincie (
    provincie_id char(2) constraint provincie_id_nn not null,
    nume_provincie varchar2(15),
    tara_id char(2),
    constraint provincie_id_pk PRIMARY KEY (provincie_id)
);

ALTER TABLE provincie
ADD (CONSTRAINT provincie_tara_fk FOREIGN KEY(tara_id) REFERENCES tara(tara_id));


CREATE TABLE locatie (
    locatie_id number(4),
    provincie_id char(2),
    oras varchar2(30) CONSTRAINT loc_oras_nn NOT NULL,
    strada varchar2(30),
    nr_strada number,
    cod_postal varchar2(10)
);

CREATE UNIQUE INDEX loc_id_pk
ON locatie (locatie_id);

alter table locatie
add (constraint loc_id_pk primary key (locatie_id),
constraint loc_prov_id_fk foreign key (provincie_id) references provincie(provincie_id),
constraint cod_pos_unq unique(cod_postal));
```

Script Output:

```
Table PROVINCIE created.

Table PROVINCIE altered.

Table LOCATIE created.

INDEX LOC_ID_PK created.

Table LOCATIE altered.
```

SQL Worksheet History project_an2

Worksheet Query Builder

```

CREATE TABLE proprietar (
    proprietar_id number(3) constraint proprietar_id_nn not null,
    nume_proprietar varchar2(20),
    prenume_proprietar varchar2(20),
    email_proprietar varchar2(30)
);

alter table proprietar
add (constraint proprie_id_pk primary key (proprietar_id));


CREATE TABLE client (
    client_id number constraint client_id_nn not null,
    nume_client varchar2(20),
    prenume_client varchar2(20),
    cnp_client number(13) constraint cnp_client_ung unique,
    nr_tel_client varchar2(10)
);

alter table client
add (constraint client_id_pk primary key (client_id));

```

Script Output X | Task completed in 0.683 seconds

Table PROPRIETAR created.

Table PROPRIETAR altered.

Table CLIENT created.

Table CLIENT altered.

SQL Worksheet History project_an2

Worksheet Query Builder

```

CREATE TABLE reprezentanta (
    reprezentanta_id number constraint reprezentanta_id_nn not null,
    nume_reprezentanta varchar2(30),
    adresa_reprezentanta varchar2(30),
    contact_reprezentanta varchar2(10)
);

alter table reprezentanta
add (constraint reprezentanta_id_pk primary key (reprezentanta_id));


CREATE TABLE angajat (
    angajat_id number constraint angajat_id_nn not null,
    nume_ang varchar2(20),
    prenume_ang varchar2(20),
    nr_tel_ang varchar2(10),
    email_ang varchar2(30) constraint email_ang_ung unique,
    cnp_ang varchar2(30) constraint cnp_ang_ung unique,
    varsta_ang number
);

CREATE UNIQUE INDEX angajat_id_pk
ON angajat (angajat_id);

alter table angajat
add (constraint angajat_id_pk primary key (angajat_id));

```

Script Output X | Task completed in 0.683 seconds

Table REPREZENTANTA created.

Table REPREZENTANTA altered.

Table ANGAJAT created.

INDEX ANGAJAT_ID_PK created.

Table ANGAJAT altered.

create.sql incercari.sql exerciti_proiect.sql ramasite.sql

SQL Worksheet History project_an2

Worksheet Query Builder

```
create table job (
    job_id varchar2(10) constraint job_id_nn not null,
    denumire_job varchar2(30) constraint den_job_nn not null,
    salariu_min number,
    salariu_max number
);

CREATE UNIQUE INDEX job_id_pk
ON job (job_id);

ALTER TABLE job
ADD (CONSTRAINT job_id_pk PRIMARY KEY(job_id));
```

Script Output Task completed in 0.683 seconds

Table JOB created.

INDEX JOB_ID_PK created.

Table JOB altered.

create.sql incercari.sql exerciti_proiect.sql ramasite.sql

SQL Worksheet History project_an2

Worksheet Query Builder

```
create table vehicul (
    vehicul_id number constraint vehicul_id_nn not null,
    client_id number,
    tip_vehicul varchar2(20) constraint tip_veh_nn not null,
    marca varchar2(20),
    an_fabricatie number(4),
    culoare varchar2(20),
    cai_putere number,
    reprezentanta_id number
);

alter table vehicul
add (constraint vehicul_id_pk primary key (vehicul_id),
constraint veh_client_id_fk foreign key(client_id) references client(client_id),
constraint vehic_tip_pk unique (vehicul_id, tip_vehicul),
constraint veh_tip_ck check (lower(tip_vehicul) in ('masina', 'autobuz', 'tir')),
constraint veh_reprezentanta_id_fk foreign key(reprezentanta_id) references reprezentanta(reprezentanta_id)
);
```

Script Output Task completed in 0.683 seconds

Table VEHICUL created.

Table VEHICUL altered.

create.sql * incercari.sql * exercitii_project.sql * ramasite.sql

SQL Worksheet History project_an2

Worksheet Query Builder

```

create table masina (
    vehicul_id number constraint veh_mas_id_pk primary key,
    tip_vehicul varchar2(20) default 'masina' constraint tip_mas_ck check (lower(tip_vehicul) = 'masina'),
    model varchar2(20),
    nr_usi number(1),
    tractiune varchar2(15),
    transmisie varchar2(15)
);

alter table masina
add (constraint veh_mas_fk foreign key (vehicul_id)
references vehicul(vehicul_id));

-----


create table autobuz (
    vehicul_id number constraint veh_bus_id_pk primary key,
    tip_vehicul varchar2(20) default 'autobuz' constraint tip_bus_ck check (lower(tip_vehicul) = 'autobuz'),
    nr_locuri number(3),
    categorie varchar2(15)
);

alter table autobuz
add (constraint veh_bus_fk foreign key (vehicul_id)
references vehicul(vehicul_id));

```

Script Output X | Task completed in 0.683 seconds

Table MASINA created.

Table MASINA altered.

Table AUTOBUZ created.

Table AUTOBUZ altered.

create.sql * incercari.sql * exercitii_project.sql * ramasite.sql

SQL Worksheet History project_an2

Worksheet Query Builder

```

create table tir (
    vehicul_id number constraint veh_tir_id_pk primary key,
    tip_vehicul varchar2(20) default 'tir' constraint tip_tir_ck check (lower(tip_vehicul) = 'tir'),
    capacitate_tone number(3)
);

alter table tir
add (constraint veh_tir_fk foreign key (vehicul_id)
references vehicul(vehicul_id));

-----


create table depozit(
    depozit_id number(4),
    denumire_depozit varchar2(30) constraint depo_den_nn not null,
    capacitate_depozit number,
    proprietar_id number(3),
    locatie_id number(4)
);

CREATE UNIQUE INDEX depo_id_pk
ON depozit (depozit_id);

alter table depozit
add (constraint depo_id_pk primary key(depozit_id),
constraint depo_prop_fk foreign key(proprietar_id) references proprietar(proprietar_id),
constraint depo_loc_fk foreign key(locatie_id) references locatie(locatie_id),
constraint depo_loc_unq unique(locatie_id));

```

Script Output X | Task completed in 0.683 seconds

Table TIR created.

Table TIR altered.

Table DEPOZIT created.

INDEX DEPO_ID_PK created.

Table DEPOZIT altered.

create.sql incercari.sql exercitii_proiect.sql ramasite.sql

SQL Worksheet History

Worksheet Query Builder

```
create table serviciu (
    serviciu_id number(4),
    depozit_id number(4),
    vehicul_id number,
    denumire_serviciu varchar2(30) constraint serv_den_nn not null,
    data_introducere date,
    data_iesire date
);

CREATE UNIQUE INDEX serv_id_pk
ON serviciu (serviciu_id) ;

alter table serviciu
add (constraint serv_id_pk primary key(serviciu_id),
constraint serv_dep_fk foreign key(depozit_id) references depozit(depozit_id),
constraint serv_veh_fk foreign key(vehicul_id) references vehicul(vehicul_id),
constraint serv_interval check (data_introducere <= data_iesire) );
```

Script Output X

Task completed in 0.683 seconds

Table SERVICIU created.

INDEX SERV_ID_PK created.

Table SERVICIU altered.

create.sql incercari.sql exercitii_proiect.sql ramasite.sql

SQL Worksheet History

Worksheet Query Builder

```
create table lucreaza (
    angajare_id number constraint angj_id_nn not null,
    depozit_id number(4) constraint angj_depo_id_nn not null,
    angajat_id number constraint angj_ang_id_nn not null,
    job_id varchar2(10) constraint angj_job_id_nn not null,
    data_angajare date constraint angj_data_nn not null,
    salar number
);

CREATE UNIQUE INDEX angj_id_pk
ON lucreaza(angajare_id);

alter table lucreaza
add (constraint angj_id_pk primary key (angajare_id),
constraint angj_depo_id_fk foreign key (depozit_id)
references depozit(depozit_id),
constraint angj_ang_id_fk foreign key (angajat_id)
references angajat(angajat_id),
constraint angj_job_id_fk foreign key (job_id)
references job(job_id)
);

commit;
```

Script Output X

Task completed in 0.683 seconds

Table LUCREAZA created.

INDEX ANGJ_ID_PK created.

Table LUCREAZA altered.

Commit complete.

Cerinta 5.

Adăugați informații coerente în tabelele create.

```
insert into regiune
values (1, 'Europa');
insert into regiune
values (2, 'Asia');
insert into regiune
values (3, 'Africa');
insert into regiune
values (4, 'America Nord');
insert into regiune
values (5, 'America Sud');
insert into regiune
values (6, 'Australia');
```

```
insert into tara
values ('RO', 'Romania', 1);
insert into tara
values ('IT', 'Italia', 1);
insert into tara
values ('BR', 'Brazilia', 5);
insert into tara
values ('EG', 'Egipt', 3);
insert into tara
values ('JP', 'Japonia', 2);
insert into tara
values ('CA', 'Canada', 4);
```

```
insert into provincie
values('MM', 'Maramures', 'RO');
insert into provincie
values('B', 'Bucuresti', 'RO');
insert into provincie
values('CJ', 'Cluj', 'RO');
insert into provincie
values('AB', 'Abruzzo', 'IT');
insert into provincie
values('ER', 'Emilia-Romagna', 'IT');
insert into provincie
values('HO', 'Hokkaido', 'JP');
insert into provincie
values('CH', 'Chubu', 'JP');
insert into provincie
values('GO', 'Goias', 'BR');
```

```
CREATE SEQUENCE locatie_seq
START WITH 1000
INCREMENT BY 10
MAXVALUE 9990
NOCACHE
NOCYCLE;

insert into locatie
values(locatie_seq.nextval, 'MM', 'Baia Mare', 'Independentei', 5, 430071 );
insert into locatie
values(locatie_seq.nextval, 'B', 'Bucuresti', 'Traian', 16, 232035 );
insert into locatie
values(locatie_seq.nextval, 'ER', 'Parma', 'Giuseppe', 261, 625910 );
insert into locatie
values(locatie_seq.nextval, 'GO', 'Rio Verde', 'Augusta Bastos', 51, 165327 );
insert into locatie
values(locatie_seq.nextval, 'HO', 'Biei', 'Chome', 178, 452681 );
insert into locatie
```

```
values(locatie_seq.nextval, 'CH', 'Chiba', 'Wangan', 21, 912376 );
insert into locatie
values(locatie_seq.nextval, 'CJ', 'Turda', 'Razboieni', 17, 726812 );
insert into locatie
values(locatie_seq.nextval, 'CJ', 'Apahida', 'Parcului', 73, 149268 );
```

```
CREATE SEQUENCE proprietar_seq
START WITH 100
INCREMENT BY 1
MAXVALUE 999
NOCACHE
NOCYCLE;

insert into proprietar
values(proprietar_seq.nextval,'Popescu', 'Ana','ana.popescu@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Sorescu','Maria','maria.sorescu@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Goja','Andrei','andrei.goja@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Albu','Sorin','sorin.albu@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Munteanu','Carla','carla.munteanu@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Matache','Ariana','ariana.matache@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Onisa','Matei','matei.onisa@gmail.com ');
insert into proprietar
values(proprietar_seq.nextval, 'Tintas','Elena','elena.tintas@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Hotea','Cosmin','cosmin.hotea@gmail.com');
insert into proprietar
values(proprietar_seq.nextval,'Turda','Petru','petru.turda@gmail.com');
```

```
CREATE SEQUENCE client_seq
START WITH 100
INCREMENT BY 10
MAXVALUE 9990
NOCACHE
NOCYCLE;

INSERT INTO client
VALUES (client_seq.nextval, 'Popescu', 'Ema', 1890112345678 , '0721123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Ionescu', 'Iris', 2900223456789 , '0732123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Stoica', 'Isabella', 3910334567890 , '0743123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Popa', 'Julia', 4920445678901 , '0754123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Dumitru', 'Lydia', 5930556789012 , '0765123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Stan', 'Maia', 6940667890123 , '0776123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Vasilescu', 'Carol', 7950778901234 , '0787123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Mihai', 'Iacob', 8960889012345 , '0798123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Georgescu', 'Oliver', 9970990123456 , '0809123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Andrei', 'Darius', 0981001234567 , '0819123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Radu', 'Felix', 1991112345678 , '0829123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Diaconu', 'Iustin', 2001223456789 , '0839123456');
```

```
CREATE SEQUENCE reprezentanta_seq
START WITH 100
INCREMENT BY 1
MAXVALUE 9990
NOCACHE
NOCYCLE;
```

```
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Tiriac Auto', 'Strada Parcului 55-57', '0728394622');
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Renaut Di-Bas', '?oseaua Fundeni 260', '0728344627');
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Das Auto Trader', 'Str. Ion Maiorescu 65', '0728394662');
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Nissan Auto', 'Strada Drumetului 26A', '0728394812');
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Toyota Motors', 'Bulevardul Pipera 16A', '0729994622');
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Opel Radacini', 'Calea Vitan 112', '0739694622');
INSERT INTO reprezentanta
VALUES (reprezentanta_seq.nextval, 'Mercedes Autoklass', 'Splaiul Unirii 166A', '0728355522');
```

```
CREATE SEQUENCE angajat_seq
START WITH 100
INCREMENT BY 1
MAXVALUE 9990
NOCACHE
NOCYCLE;
```

```
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Dragos', 'Ana', '0722345678', 'ana.dragos@gmail.com', '1890223456789', 20);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Constantinescu', 'Maria', '0733345678', 'maria.constantinescu@gmail.com',
'2900334567890', 25);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Manescu', 'Elena', '0744345678', 'elena.manescu@gmail.com', '3910445678901', 30);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Petrescu', 'Ioana', '0755345678', 'ioana.petrescu@gmail.com', '4920556789012', 35);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Balan', 'Andreea', '0766345678', 'andreea.balan@gmail.com', '5930667890123', 40);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Dobreanu', 'Sara', '0777345678', 'sara.dobreanu@gmail.com', '6940778901234', 45);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Munteanu', 'Alex', '0788345678', 'alex.munteanu@gmail.com', '7950889012345', 50);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Stanciu', 'Matei', '0799345678', 'matei.stanciu@gmail.com', '8960990123456', 55);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Grigorescu', 'David', '0800345678', 'david.grigorescu@gmail.com', '9970001234567',
60);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Florescu', 'Gabriel', '0811345678', 'gabriel.florescu@gmail.com', '0981112345678', 28);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Preda', 'Mihai', '0822345678', 'mihai.preda@gmail.com', '1991223456789', 37);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Mocanu', 'Andrei', '0833345678', 'andrei.mocanu@gmail.com', '2001334567890', 42);

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Iordanescu', 'Giulia', '0721123456', 'giulia.iordanescu@gmail.com', '3011445678901',
32);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Popescu', 'Sofia', '0765123456', 'sofia.popescu@gmail.com', '4021556789012', 35);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Vladescu', 'Isabella', '0742123456', 'isabella.vladescu@gmail.com', '5031667890123',
41);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Manescu', 'Alessandro', '0753123456', 'alessandro.manescu@gmail.com',
'6041778901234', 45);
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Florescu', 'Francesco', '0734123456', 'francesco.florescu@gmail.com',
'7051889012345', 48);
```

```
INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Stefanescu', 'Leo', '0786123456', 'leo.stefanescu@gmail.com', '8061990123456', 50);
```

```
INSERT INTO job
VALUES ('MEC_AUT', 'Mecanic auto', 2100 , 5000);
INSERT INTO job
VALUES ('INS_TEH', 'Inspector tehnic', 2500 , 5500);
INSERT INTO job
VALUES ('OP_STC', 'Operator de stocare', 2900 , 6000);
INSERT INTO job
VALUES ('SOF_LIV', 'Sofer de livrare', 3300 , 6500);
INSERT INTO job
VALUES ('TEH_VOP', 'Tehnician de vopsire', 3700 , 7000);
INSERT INTO job
VALUES ('PAZ', 'Paznic', 4100 , 7500);
INSERT INTO job
VALUES ('CON', 'Contabil', 4500 , 8000);
```

```
CREATE SEQUENCE vehicul_seq
START WITH 100
INCREMENT BY 10
MAXVALUE 9990
NOCACHE
NOCYCLE;

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,100, 'masina' , 'Mercedes-Benz' , 2004 , 'Alb' , 120, 106);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,110, 'autobuz' , 'Volvo' , 2008 , 'Negru' , 150, 103);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,110, 'masina' , 'Volkswagen' , 2001 , 'Argintiu' , 180, 102);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,120, 'tir' , 'Mercedes-Benz' , 2019 , 'Rosu' , 200, 100);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,130, 'masina' , 'Iveco' , 2007 , 'Albastru' , 240, 101);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,130, 'tir' , 'Daimler' , 2015 , 'Verde' , 260, 104);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,140, 'autobuz' , 'Renault' , 2010 , 'Galben' , 280, 105);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,150, 'autobuz' , 'Ford' , 2002 , 'Portocaliu' , 290, 103);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,160, 'tir' , 'Volkswagen' , 2018 , 'Gri' , 250, 105);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,170, 'masina' , 'Fiat' , 1999 , 'Maro' , 210, 106);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,180, 'tir' , 'Isuzu' , 2012 , 'Alb' , 170, 104);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,180, 'autobuz' , 'Hino' , 2006 , 'Negru' , 140, 101);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,190, 'masina' , 'Ford' , 2003 , 'Argintiu' , 230, 102);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,200, 'autobuz' , 'Mercedes-Benz' , 2016 , 'Verde' , 270, 100);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,210, 'masina' , 'Toyota' , 2011 , 'Negru' , 300, 104);
INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,210, 'tir' , 'Man' , 2010 , 'Argintiu' , 350, 106);
```

```
insert into masina
values(100, 'S Class', 4, 'fata', 'manuala');
insert into masina
values(120, 'Golf 3', 4, 'fata', 'manuala');
insert into masina
values(140, 'Massif 4', 2, 'fata', 'manuala');
insert into masina
values(190, 'Tipo', 4, 'spate', 'manuala');
```

```

insert into masina
values(220, 'Focus', 4, 'spate', 'manuala');
insert into masina
values(240,'Corola', 4, 'fata', 'automata');

insert into autobuz
values(110, 60, 'turistic');
insert into autobuz
values(160,30,'urban');
insert into autobuz
values(170,20,'scolar');
insert into autobuz
values(210,50, 'turistic');
insert into autobuz
values(230, 25, 'urban');

insert into tir
values (130,10);
insert into tir
values (150,15);
insert into tir
values (180,40);
insert into tir
values (200,25);
insert into tir
values (250,30);

-----
CREATE SEQUENCE depozit_seq
START WITH 1000
INCREMENT BY 100
MAXVALUE 9990
NOCACHE
NOCYCLE;

ALTER TABLE depozit
MODIFY denumire_depozit varchar2(100);

INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Depozitul AutoExpert', 1200, 100, 1000);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Centrul Logistic AutoMax', 800, 101, 1010);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Depozitul Integrat AutoPro', 1000, 102, 1020);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Spatiul de Stocare VehiPark', 600, 103, 1030);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Depozitul Avansat AutoTech', 900, 106, 1060);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Complexul de Depozitare AutoPrime', 1100, 105, 1050);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Depozitul Specializat AutoExcel', 1300, 100, 1040);
INSERT INTO depozit
VALUES (depozit_seq.nextval, 'Centrul de Distributie AutoElite', 700, 107, 1070);

-----
CREATE SEQUENCE serviciu_seq
START WITH 1000
INCREMENT BY 1
MAXVALUE 9990
NOCACHE
NOCYCLE;

INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1000, 100, 'vopsire', to_date('17-MAY-2022','dd-mon-yyyy'),
to_date('20-MAY-2022','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1000, 120, 'schimb ulei', to_date('20-JAN-2021','dd-mon-yyyy'),
to_date('22-JAN-2021','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1000, 170, 'geometrie roti', to_date('30-APR-2022','dd-mon-yyyy'),

```

```

to_date('02-MAY-2022','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1100, 110, 'depozitare', to_date('20-DEC-2021','dd-mon-yyyy'),
to_date('05-JAN-2022','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1100, 130, 'revizie tehnica', to_date('13-SEP-2021','dd-mon-yyyy'),
to_date('16-SEP-2021','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1100, 160, 'curatare', to_date('17-MAR-2020','dd-mon-yyyy'),
to_date('21-MAR-2020','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1300, 140, 'polish', to_date('10-SEP-2022','dd-mon-yyyy'),
to_date('13-SEP-2022','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1300, 170, 'revizie tehnica', to_date('04-NOV-2021','dd-mon-yyyy'),
to_date('10-NOV-2021','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1400, 150, 'depozitare', to_date('16-JUN-2022','dd-mon-yyyy'),
to_date('16-JUL-2022','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1400, 110, 'reparatie parbriz', to_date('09-SEP-2020','dd-mon-yyyy'),
to_date('14-SEP-2020','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1500, 180, 'depozitare', to_date('01-FEB-2021','dd-mon-yyyy'),
to_date('31-AUG-2021','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1500, 190, 'curatare chimica', to_date('20-NOV-2022','dd-mon-yyyy'),
to_date('29-NOV-2022','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1600, 210, 'schimbare placute frana', to_date('10-MAY-2023','dd-mon-yyyy'),
to_date('13-MAY-2023','dd-mon-yyyy'));
INSERT INTO serviciu
VALUES (serviciu_seq.nextval, 1700, 220, 'schimb ulei', to_date('20-FEB-2023','dd-mon-yyyy'),
to_date('20-FEB-2023','dd-mon-yyyy'));

-----
```

```
CREATE SEQUENCE lucreaza_seq
```

```

START WITH 100
INCREMENT BY 11
MAXVALUE 9990
NOCACHE
NOCYCLE;
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 100, 'MEC_AUT', to_date('01-01-2008','dd-mm-yyyy'),2200);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 101, 'INS_TEH', to_date('15-05-2010','dd-mm-yyyy'),2600);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 102, 'OP_STC', to_date('28-09-2012','dd-mm-yyyy'),3000);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 103, 'SOF_LIV', to_date('10-12-2014','dd-mm-yyyy'),3400);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 104, 'TEH_VOP', to_date('22-08-2016','dd-mm-yyyy'),3800);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 105, 'PAZ', to_date('05-11-2019','dd-mm-yyyy'),4200);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1000, 106, 'CON', to_date('17-03-2021','dd-mm-yyyy'),4600);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1100, 107, 'MEC_AUT', to_date('01-01-2020','dd-mm-yyyy'),2600);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1100, 108, 'INS_TEH', to_date('15-06-2021','dd-mm-yyyy'),3000);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1100, 109, 'OP_STC', to_date('30-12-2022','dd-mm-yyyy'),3400);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1100, 110, 'SOF_LIV', to_date('05-02-2020','dd-mm-yyyy'),3800);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1100, 111, 'TEH_VOP', to_date('10-09-2021','dd-mm-yyyy'),4200);
```

```
INSERT INTO lucreaza
```

```
VALUES (lucreaza_seq.nextval, 1100, 112, 'PAZ', to_date('25-11-2022','dd-mm-yyyy'),4600);
```

```

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 113, 'CON', to_date('10-03-2020','dd-mm-yyyy'),5000);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 114, 'MEC_AUT', to_date('05-02-2018','dd-mm-yyyy'),3210);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 115, 'INS_TEH', to_date('18-06-2018','dd-mm-yyyy'),3610);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 116, 'OP_STC', to_date('09-09-2019','dd-mm-yyyy'),4010);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 117, 'SOF_LIV', to_date('22-12-2019','dd-mm-yyyy'),4410);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 100, 'TEH_VOP', to_date('14-05-2020','dd-mm-yyyy'),4810);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 101, 'PAZ', to_date('27-08-2020','dd-mm-yyyy'),5210);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 102, 'CON', to_date('10-11-2020','dd-mm-yyyy'),5610);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 103, 'MEC_AUT', to_date('15-03-2016','dd-mm-yyyy'),2850);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 104, 'INS_TEH', to_date('22-09-2017','dd-mm-yyyy'),3250);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 105, 'OP_STC', to_date('07-05-2018','dd-mm-yyyy'),3650);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 106, 'SOF_LIV', to_date('18-11-2019','dd-mm-yyyy'),4050);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 107, 'TEH_VOP', to_date('03-07-2020','dd-mm-yyyy'),4450);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 108, 'PAZ', to_date('12-01-2021','dd-mm-yyyy'),4850);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1300, 109, 'CON', to_date('27-08-2022','dd-mm-yyyy'),5250);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 110, 'MEC_AUT', to_date('10-08-2016','dd-mm-yyyy'),4500);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 111, 'INS_TEH', to_date('05-04-2017','dd-mm-yyyy'),4900);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 112, 'OP_STC', to_date('21-12-2018','dd-mm-yyyy'),5300);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 113, 'SOF_LIV', to_date('16-10-2019','dd-mm-yyyy'),5700);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 114, 'TEH_VOP', to_date('09-06-2020','dd-mm-yyyy'),6100);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 115, 'PAZ', to_date('25-02-2021','dd-mm-yyyy'),6500);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1400, 116, 'CON', to_date('14-09-2022','dd-mm-yyyy'),6900);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1500, 117, 'MEC_AUT', to_date('01-02-2023','dd-mm-yyyy'),4050);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1500, 101, 'INS_TEH', to_date('15-03-2023','dd-mm-yyyy'),4450);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1500, 105, 'OP_STC', to_date('10-04-2023','dd-mm-yyyy'),4850);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1600, 110, 'SOF_LIV', to_date('01-02-2023','dd-mm-yyyy'),4050);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1600, 115, 'TEH_VOP', to_date('15-03-2023','dd-mm-yyyy'),4450);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1600, 105, 'PAZ', to_date('10-04-2023','dd-mm-yyyy'),4850);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1700, 108, 'CON', to_date('11-11-2021','dd-mm-yyyy'),6250);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1700, 106, 'PAZ', to_date('16-05-2022','dd-mm-yyyy'),5800);

commit;

```

Urmeaza printscreen-uri doveditoare cu faptul ca informatiile au fost adaugate cu succes.

The screenshot shows the MySQL Workbench interface with several tabs at the top: 'creare.sql', 'incercari.sql', and 'exercitii_proiect.s'. The 'Script Output' tab is active, displaying the results of an SQL script. The script inserts data into three tables: 'regiune', 'tara', and 'provincie'. The 'regiune' table has 6 rows. The 'tara' table has 8 rows. The 'provincie' table has 10 rows. Each insert statement is followed by a confirmation message '1 row inserted.'.

```
insert into regiune
values (1, 'Europa');
1 row inserted.

insert into regiune
values (2, 'Asia');
1 row inserted.

insert into regiune
values (3, 'Africa');
1 row inserted.

insert into regiune
values (4, 'America Nord');
1 row inserted.

insert into regiune
values (5, 'America Sud');
1 row inserted.

insert into regiune
values (6, 'Australia');

----- 1 row inserted.

insert into tara
values ('RO', 'Romania', 1);
1 row inserted.

insert into tara
values ('IT', 'Italia', 1);
1 row inserted.

insert into tara
values ('BR', 'Brasilia', 5);
1 row inserted.

insert into tara
values ('EG', 'Egipt', 3);
1 row inserted.

insert into tara
values ('JP', 'Japonia', 2);
1 row inserted.

insert into tara
values ('CA', 'Canada', 4);

----- 1 row inserted.

insert into provincie
values('MM', 'Maramures', 'RO');
1 row inserted.

insert into provincie
values('B', 'Bucuresti', 'RO');
1 row inserted.

insert into provincie
values('CJ', 'Cluj', 'RO');
1 row inserted.

insert into provincie
values('AB', 'Abruzzo', 'IT');
1 row inserted.

insert into provincie
values('ER', 'Emilia-Romagna', 'IT');
1 row inserted.

insert into provincie
values('HO', 'Hokkaido', 'JP');
1 row inserted.

insert into provincie
values('CH', 'Chubu', 'JP');
1 row inserted.

insert into provincie
values('GO', 'Goiias', 'BR');
1 row inserted.

----- 1 row inserted.

----- 1 row inserted.
```

SQL Worksheet History | Task completed in 0.253 seconds

Worksheet Query Builder

```
CREATE SEQUENCE client_seq
START WITH    100
INCREMENT BY  10
MAXVALUE     9990
NOCACHE
NOCYCLE;

INSERT INTO client
VALUES (client_seq.nextval, 'Popescu', 'Ema', 1880112345678 , '0721123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Ionescu', 'Iris', 2900223456789 , '0732123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Stoica', 'Isabella', 3910334567890 , '0743123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Popa', 'Julia', 4920445678901 , '0754123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Dumitru', 'Lydia', 5930556789012 , '0765123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Stan', 'Maiu', 6940667890123 , '0776123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Vasilescu', 'Carol', 7950778901234 , '0787123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Mihai', 'Iacob', 8960889012345 , '0798123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Georgescu', 'Oliver', 9970990123456 , '0809123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Andrei', 'Darius', 0981001234567 , '0819123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Radu', 'Felini', 1991112345678 , '0829123456');
INSERT INTO client
VALUES (client_seq.nextval, 'Diaconu', 'Iustin', 2001223456789 , '0839123456');

-----


CREATE SEQUENCE representanta_seq
START WITH    100
INCREMENT BY  1
MAXVALUE     9990
NOCACHE
NOCYCLE;

INSERT INTO representanta
VALUES (representanta_seq.nextval, 'Tiriac Auto', 'Strada Parcului 55-57' , '0728294622');
INSERT INTO representanta
VALUES (representanta_seq.nextval, 'Renaut Di-Bas', '?oseaua Fundeni 260' , '0728344627');
INSERT INTO representanta
VALUES (representanta_seq.nextval, 'Das Auto Trader', 'Str. Ion Maiorescu 65' , '0728394628');
INSERT INTO representanta
```

Script Output X | Task completed in 0.253 seconds

Sequence CLIENT_SEQ created.

1 row inserted.

Sequence REPREZENTANTA_SEQ created.

1 row inserted.

SQL Worksheet History | Task completed in 0.17 seconds

Sequence ANGAJAT_SEQ created.

```

CREATE SEQUENCE angajat_seq
START WITH    100
INCREMENT BY  1
MAXVALUE     9990
NOCACHE
NOCYCLE;

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Dragos', 'Ana', '0722345678', 'ana.dragos@gmail.com', '1890223456789', 20);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Constantinescu', 'Maria', '0733345678', 'maria.constantinescu@gmail.com', '29002234567890', 25);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Manescu', 'Elena', '0744345678', 'elena.manescu@gmail.com', '3910445678901', 30);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Petrescu', 'Ioana', '0755345678', 'ioana.petrescu@gmail.com', '4920556789012', 35);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Balan', 'Andreea', '0766345678', 'andreea.balan@gmail.com', '5930667890123', 40);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Dobreanu', 'Sara', '0777345678', 'sara.dobreanu@gmail.com', '6940778901234', 45);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Munteanu', 'Alex', '0788345678', 'alex.munteanu@gmail.com', '7950889012345', 50);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Stanciu', 'Matei', '0799345678', 'matei.stanciu@gmail.com', '8960990123456', 55);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Grigorescu', 'David', '0800345678', 'david.grigorescu@gmail.com', '9970001234567', 60);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Florescu', 'Gabriel', '0811345678', 'gabriel.florescu@gmail.com', '0981112345678', 28);
1 row inserted.

VALUES (angajat_seq.nextval, 'Preda', 'Mihai', '0822345678', 'mihai.preda@gmail.com', '1991223456789', 37);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Mocanu', 'Andrei', '0833345678', 'andrei.mocanu@gmail.com', '20012234567890', 42);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Iordanescu', 'Giulia', '0731123456', 'giulia.iordanescu@gmail.com', '3011445678901', 32);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Popescu', 'Sofia', '0765123456', 'sofia.popescu@gmail.com', '4021556789012', 35);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Vlădescu', 'Isabella', '0742123456', 'isabella.vladescu@gmail.com', '5031667890123', 41);
1 row inserted.

INSERT INTO angajat
VALUES (angajat_seq.nextval, 'Manescu', 'Alessandro', '0752123456', 'alessandro.manescu@gmail.com', '6041778901234', 45);
1 row inserted.

VALUES (angajat_seq.nextval, 'Florescu', 'Francesco', '0734123456', 'francesco.florescu@gmail.com', '7051889012345', 48);
1 row inserted.

VALUES (angajat_seq.nextval, 'Stefanescu', 'Leo', '0786123456', 'leo.stefanescu@gmail.com', '8061990123456', 50);
1 row inserted.

-----
```

1 row inserted.

Script Output X | Task completed in 0.17 seconds

1 row inserted.

SQL Worksheet History | Task completed in 0.18 seconds

Sequence VEHICUL_SEQ created.

```

INSERT INTO job
VALUES ('MEC_AUT', 'Mecanic auto', 2100 , 5000);
1 row inserted.

INSERT INTO job
VALUES ('INS_TEH', 'Inspector tehnic', 2500 , 5500);
1 row inserted.

INSERT INTO job
VALUES ('OP_STC', 'Operator de stocare', 2900 , 6000);
1 row inserted.

INSERT INTO job
VALUES ('SOF_LIV', 'Sofer de livrare', 3300 , 6500);
1 row inserted.

INSERT INTO job
VALUES ('TEH_VOP', 'Technician de vopsire', 3700 , 7000);
1 row inserted.

INSERT INTO job
VALUES ('PAZ', 'Paznic', 4100 , 7500);
1 row inserted.

INSERT INTO job
VALUES ('CON', 'Contabil', 4500 , 8000);
1 row inserted.

-----
```

```

CREATE SEQUENCE vehicul_seq
START WITH    100
INCREMENT BY  10
MAXVALUE     9990
NOCACHE
NOCYCLE;

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,100, 'masina', 'Mercedes-Benz', 2004 , 'Alb' , 120, 106);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,110, 'autobus', 'Volvo' , 2008 , 'Negru' , 150, 108);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,110, 'masina', 'Volkswagen' , 2001 , 'Argintiu' , 180, 102);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,120, 'tir' , 'Mercedes-Benz' , 2019 , 'Rosu' , 200, 100);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,130, 'masina', 'Iveco' , 2007 , 'Albastru' , 240, 101);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,130, 'tir' , 'Daimler' , 2015 , 'Verde' , 260, 104);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,140, 'autobus', 'Renault' , 2010 , 'Galben' , 280, 105);
1 row inserted.

INSERT INTO vehicul
VALUES (vehicul_seq.nextval ,150, 'autobus', 'Ford' , 2002 , 'Portocaliu' , 290, 103);
1 row inserted.

VALUES (vehicul_seq.nextval ,160, 'tir' , 'Volkswagen' , 2018 , 'Gri' , 250, 105);
1 row inserted.

VALUES (vehicul_seq.nextval ,170, 'masina', 'Fiat' , 1999 , 'Maro' , 210, 106);
1 row inserted.

VALUES (vehicul_seq.nextval ,180, 'tir' , 'Isuzu' , 2012 , 'Alb' , 170, 104);
1 row inserted.

VALUES (vehicul_seq.nextval ,180, 'autobus', 'Hino' , 2006 , 'Negru' , 140, 101);
1 row inserted.
```

1 row inserted.

Script Output X | Task completed in 0.18 seconds

```
insert into masina
values(100, 'S Class', 4, 'fata', 'manuala');
insert into masina
values(120, 'Golf 3', 4, 'fata', 'manuala');
insert into masina
values(140, 'Massif 4', 2, 'fata', 'manuala');
insert into masina
values(160, 'Tipo', 4, 'spate', 'manuala');
insert into masina
values(220, 'Focus', 4, 'spate', 'manuala');
insert into masina
values(240,'Corola', 4, 'fata', 'automata');

insert into autobuz
values(110, 60, 'turistic');
insert into autobuz
values(160,30,'urban');
insert into autobuz
values(170,20,'scolar');
insert into autobuz
values(210,50,'turistic');
insert into autobuz
values(230, 25, 'urban');

insert into tir
values (130,10);
insert into tir
values (150,15);
insert into tir
values (180,40);
insert into tir
values (200,25);
insert into tir
values (250,30);
```

```
CREATE SEQUENCE deposit_seq
START WITH      1000
INCREMENT BY   100
MAXVALUE       9990
NOCACHE
NOCYCLE;

ALTER TABLE deposit
MODIFY denumire_deposit varchar2(100);

INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Depositul AutoExpert', 1200, 100, 1000);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Centrul Logistic AutoMax', 800, 101, 1010);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Depositul Integrat AutoPro', 1000, 102, 1020);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Spatiul de Stocare VehiPark', 600, 103, 1030);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Depositul Avansat AutoTech', 900, 106, 1060);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Complexul de Depositate AutoPrime', 1100, 105, 1050);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Depositul Specializat AutoExcel', 1300, 100, 1040);
INSERT INTO deposit
VALUES (deposit_seq.nextval, 'Centrul de Distributie AutoElite', 700, 107, 1070);
```

Script Output | Task completed in 0.163 seconds

```
Table DEPOSIT altered

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

```
CREATE SEQUENCE serviciu_seq
    START WITH      1000
    INCREMENT BY   1
    NOCACHE
    NOCYCLE;
    .
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1000, 100, 'vopsire', to_date('
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1000, 120, 'schimb ulei', to_date(
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1000, 170, 'geometrie roti', to_
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1100, 110, 'depositare', to_da
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1100, 130, 'revisie tehnica', to_
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1100, 160, 'curatare', to_date(
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1300, 140, 'polish', to_date(
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1300, 170, 'revisie tehnica', to_
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1400, 150, 'depositare', to_da
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1400, 110, 'reparatie parbriz', to_
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1500, 180, 'depositare', to_da
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1500, 190, 'curatare chimica', to_
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1600, 210, 'schimb placute', to_
    INSERT INTO serviciu
    VALUES (serviciu_seq.nextval, 1700, 220, 'schimb ulei', to_da
```

Sequence SERVICEID SEO created

```
- 1 row inserted  
  
1 row inserted  
  
1 row inserted  
  
1 row inserted
```

| Task completed in 0.103 seconds

```
Sequence SERVICIU_SEQ created.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

SQL Worksheet History 0.2 seconds

Worksheet Query Builder

```

CREATE SEQUENCE lucreaza_seq
START WITH    100
INCREMENT BY  11
MAXVALUE     9990
NOCACHE
NOCYCLE;

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 100, 'MEC_AUT', to_date('01-01-2008','dd-mm-yyyy'),2200);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 101, 'INS_TEH', to_date('15-05-2010','dd-mm-yyyy'),2600);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 102, 'OP_STC', to_date('28-09-2012','dd-mm-yyyy'),3000);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 103, 'SOF_LIV', to_date('10-12-2014','dd-mm-yyyy'),3400);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 104, 'TEH_VOP', to_date('22-08-2016','dd-mm-yyyy'),3800);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 105, 'PAZ', to_date('05-11-2019','dd-mm-yyyy'),4200);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1000, 106, 'CON', to_date('17-03-2021','dd-mm-yyyy'),4600);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 107, 'MEC_AUT', to_date('01-01-2020','dd-mm-yyyy'),2600);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 108, 'INS_TEH', to_date('15-06-2021','dd-mm-yyyy'),3000);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 109, 'OP_STC', to_date('30-12-2022','dd-mm-yyyy'),3400);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 110, 'SOF_LIV', to_date('05-02-2020','dd-mm-yyyy'),3800);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 111, 'TEH_VOP', to_date('10-09-2021','dd-mm-yyyy'),4200);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 112, 'PAZ', to_date('25-11-2022','dd-mm-yyyy'),4600);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1100, 113, 'CON', to_date('10-03-2020','dd-mm-yyyy'),5000);

INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 114, 'MEC_AUT', to_date('05-02-2018','dd-mm-yyyy'),3210);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 115, 'INS_TEH', to_date('18-06-2018','dd-mm-yyyy'),3610);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 116, 'OP_STC', to_date('09-09-2019','dd-mm-yyyy'),4010);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 117, 'SOF_LIV', to_date('22-12-2019','dd-mm-yyyy'),4410);
INSERT INTO lucreaza
VALUES (lucreaza_seq.nextval, 1200, 100, 'TEH_VOP', to_date('14-05-2020','dd-mm-yyyy'),4810);

```

Script Output X | Task completed in 0.2 seconds

Sequence LUCREAZA_SEQ created.

1 row inserted.

Urmeaza sa fac cateva printscreens cu select-uri ale unor tabele pentru a demonstra ca informatiile au fost inserate corect.

Worksheet Query Builder

```

select * from locatie;

```

Script Output X | Query Result X | All Rows Fetched: 8 in 0.003 seconds

LOCATIE_ID	PROVINCIE_ID	ORAS	STRADA	NR_STRADA	COD_POSTAL
1	1000 MM	Baia Mare	Independentei		5 430071
2	1010 B	Bucuresti	Traian		16 232035
3	1020 ER	Parma	Giuseppe		261 625910
4	1030 GO	Rio Verde	Augusta Bastos		51 165327
5	1040 HO	Bieei	Chome		178 452681
6	1050 CH	Chiba	Wangan		21 912376
7	1060 CJ	Turda	Razboieni		17 726812
8	1070 CJ	Apahida	Parcului		73 149268

Worksheet | Query Builder

```
select * from depozit;
```

Script Output | Query Result | All Rows Fetched: 8 in 0.016 seconds

DEPOZIT_ID	DENUMIRE_DEPOZIT	CAPACITATE_DEPOZIT	PROPRIETAR_ID	LOCATIE_ID
1	1000 Depositul AutoExpert	1200	100	1000
2	1100 Centrul Logistic AutoMax	800	101	1010
3	1200 Depozitul Integrat AutoPro	1000	102	1020
4	1300 Spatiul de Stocare VehiPark	600	103	1030
5	1400 Depozitul Avansat AutoTech	900	106	1060
6	1500 Complexul de Depozitare AutoPrime	1100	105	1050
7	1600 Depositul Specializat AutoExcel	1300	100	1040
8	1700 Centrul de Distributie AutoElite	700	107	1070

create.sql | incercari.sql | exerciti_proiect.sql | ramasite.sql | SQL Worksheet | History

Worksheet | Query Builder

```
select * from lucreaza;
```

Script Output | Query Result | All Rows Fetched: 43 in 0.015 seconds

ANGAJARE_ID	DEPOZIT_ID	ANGAJAT_ID	JOB_ID	DATA_ANGAJARE	SALAR
1	100	1000	100 MEC_AUT	01-JAN-08	2200
2	111	1000	101 INS_TEH	15-MAY-10	2600
3	122	1000	102 OP_STC	28-SEP-12	3000
4	133	1000	103 SOF_LIV	10-DEC-14	3400
5	144	1000	104 TEH_VOP	22-AUG-16	3800
6	155	1000	105 PAZ	05-NOV-19	4200
7	166	1000	106 CON	17-MAR-21	4600
8	177	1100	107 MEC_AUT	01-JAN-20	2600
9	188	1100	108 INS_TEH	15-JUN-21	3000
10	199	1100	109 OP_STC	30-DEC-22	3400
11	210	1100	110 SOF_LIV	05-FEB-20	3800
12	221	1100	111 TEH_VOP	10-SEP-21	4200

Worksheet | Query Builder

```
select * from serviciu;
```

Script Output | Query Result | All Rows Fetched: 14 in 0.001 seconds

SERVICIU_ID	DEPOZIT_ID	VEHICUL_ID	DENUMIRE_SERVICIU	DATA_INTRODUCERE	DATA_IESIRE
1	1000	1000	100 vopsire	17-MAY-22	20-MAY-22
2	1001	1000	120 schimb ulei	20-JAN-21	22-JAN-21
3	1002	1000	170 geometrie roti	30-APR-22	02-MAY-22
4	1003	1100	110 depozitare	20-DEC-21	05-JAN-22
5	1004	1100	130 revizie tehnica	13-SEP-21	16-SEP-21
6	1005	1100	160 curatare	17-MAR-20	21-MAR-20
7	1006	1300	140 polish	10-SEP-22	13-SEP-22
8	1007	1300	170 revizie tehnica	04-NOV-21	10-NOV-21
9	1008	1400	150 depozitare	16-JUN-22	16-JUL-22
10	1009	1400	110 reparatie parbriz	09-SEP-20	14-SEP-20
11	1010	1500	180 depozitare	01-FEB-21	31-AUG-21
12	1011	1500	190 curatare chimica	20-NOV-22	29-NOV-22
13	1012	1600	210 schimbare placute frana	10-MAY-23	13-MAY-23
14	1013	1700	220 schimb ulei	20-FEB-23	20-FEB-23

Cerinta 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.

Sa se stearga din tabelul SERVICIU inregistrările care au 'data de ieșire' în anul 2021 și să se retină aceste stergeri pentru a putea organiza o tombola astfel: clientii care detin aceste vehicule vor fi stocati intr-un tablou iar pentru fiecare se va afisa id-ul clientului și vehiculul cu care candideaza in tombola (un client poate candida cu mai multe vehicule). Se aleg 2 numere aleatoare și se extrag castigatorii. Se afiseaza numele și prenumele castigatorilor, id-ul lor cat și numarul de telefon la care trebuie să fie contactati pentru a-i anunta de castig. La final, inserati in tabelul SERVICIU inregistrările sterse.

```
create or replace procedure cerinta_6 as
```

```
type tablou_indexat is table of serviciu%rowtype index by binary_integer;
type tablou_imbricat is table of number;
type vector is varray(20) of client%rowtype;

tidx tablou_indexat;
timb tablou_imbricat := tablou_imbricat();
timb_dist tablou_imbricat := tablou_imbricat();
vec vector := vector();

cnt number;
client_vehicul number;
aux number;
random1 number;
random2 number;
precedent number := 0;
idx number := 1;

begin
--stergere din tabel și salvare in tablou indexat
    delete from serviciu
    where to_char(data_ieșire,'yyyy') = '2021'
    returning serviciu_id, depozit_id, vehicul_id, denumire_serviciu, data_introducere, data_ieșire
    bulk collect into tidx;

    cnt := tidx.count;
    if cnt <> 0 then
        for i in tidx.first..tidx.last loop
            select client_id
            into client_vehicul
            from vehicul
            where vehicul_id = tidx(i).vehicul_id;
--adaugare id pentru tombola in tabloul imbricat
            timb.extend;
            timb(i) := client_vehicul;
            dbms_output.put_line('Clientul ' || timb(i) || ' candideaza cu vehiculul ' || tidx(i).vehicul_id);
        end loop;

        cnt := timb.count;

--sortam timb
        for i in timb.first..timb.last loop
            for j in (i + 1)..timb.last loop
                if (timb(i) > timb(j)) then
                    aux := timb(i);
                    timb(i) := timb(j);
                    timb(j) := aux;
                end if;
            end loop;
        end loop;
    end if;
end;
```

```

        end if;
    end loop;
end loop;

cnt := timb.count+1;
idx := 1;

while idx <= timb.last loop
    if timb(idx) = precedent then
        timb.delete(idx);
        cnt := cnt - 1;
        idx := idx +1;
    else
        precedent := timb(idx);
        idx := idx +1;
    end if;
end loop;

cnt := timb.count;
idx := 1;
for i in timb.first..timb.last loop
    if timb.exists(i) then
        timb_dist.extend;
        timb_dist(idx) := timb(i);
        idx := idx+1;
    end if;
end loop;

cnt := timb_dist.count;

if cnt = 1 then
    vec.extend();

    select *
    into vec(1)
    from client
    where client_id = timb(1);
    dbms_output.put_line('Castigatorul este clientul cu id-ul ' || vec(1).client_id);
    dbms_output.put_line('Contactati clientul ' || vec(1).client_id || ' la numarul: ' || vec(1).nr_tel_client);
else
    select (trunc(dbms_random.VALUE*cnt)+1)
    into random1
    from dual;

    random2 := random1;

    while random2 = random1 loop
        select (trunc(dbms_random.VALUE*cnt)+1)
        into random2
        from dual;
    end loop;

    random1 := timb_dist(random1);
    random2 := timb_dist(random2);

    vec.extend();
    vec.extend();

--adaugam in vector informatiile despre castigatori
    select *
    into vec(1)
    from client
    where client_id = random1;

    select *

```

```

into vec(2)
from client
where client_id = random2;

dbms_output.new_line;
dbms_output.put_line('Cei doi castigatori sunt clientii cu id-urile ' || vec(1).client_id || ' si ' ||
vec(2).client_id);
    dbms_output.put_line('Contactati clientul ' || vec(1).nume_client || ' ' || vec(1).prenume_client || ' cu
id-ul ' || vec(1).client_id || ' la numarul de telefon: ' || vec(1).nr_tel_client);
    dbms_output.put_line('Contactati clientul ' || vec(2).nume_client || ' ' || vec(2).prenume_client || ' cu
id-ul ' || vec(2).client_id || ' la numarul de telefon: ' || vec(2).nr_tel_client);
end if;

for i in tidx.first..tidx.last loop
    insert into serviciu
    values tidx(i);
end loop;

else

    dbms_output.put_line('Nu exista candidati la tombola');

end if;
end cerinta_6;
/
begin
    cerinta_6;
end;
/

```

Urmeaza printscreens-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul.

```

create or replace procedure cerinta_6 as

    type tablou_indexat is table of serviciu%rowtype index by binary_integer;
    type tablou_imbricat is table of number;
    type vector is varray(20) of client%rowtype;

    tidx tablou_indexat;
    timb tablou_imbricat := tablou_imbricat();
    timb_dist tablou_imbricat := tablou_imbricat();
    vec vector := vector();

    cnt number;
    client_vehicul number;
    aux number;
    random number;
    random2 number;
    precedent number := 0;
    idx number := 1;

    begin
        --stergere din tabel si salvare in tablou indexat
        delete from serviciu
        where to_char(data_iesire,'yyyy') = '2021'
        returning serviciu_id, depozit_id, vehicul_id, denumire_serviciu, data_introducere, data_iesire
        bulk collect into tidx;

        cnt := tidx.count;
        if cnt <> 0 then
            for i in tidx.first..tidx.last loop
                select client_id
                into client_vehicul
                from vehicul
                where vehicul_id = tidx(i).vehicul_id;
                --adaugare id pentru tombola in tabloul imbricat
                timb.extend;
                timb(i) := client_vehicul;
                dbms_output.put_line('Clientul ' || timb(i) || ' candideaza cu vehiculul ' || tidx(i).vehicul_id);
            end loop;

            cnt := timb.count;
        end if;
    end;

```

Worksheet Query Builder

```

--sortam timb
for i in timb.first..timb.last loop
    for j in (i + 1)..timb.last loop
        if (timb(i) > timb(j)) then
            aux := timb(i);
            timb(i) := timb(j);
            timb(j) := aux;
        end if;
    end loop;
end loop;

cnt := timb.count+1;
idx := 1;

while idx <= timb.last loop
    if timb(idx) = precedent then
        timb.delete(idx);
        cnt := cnt - 1;
        idx := idx +1;
    else
        precedent := timb(idx);
        idx := idx +1;
    end if;
end loop;

cnt := timb.count;
idx := 1;
for i in timb.first..timb.last loop
    if timb.exists(i) then
        timb_dist.extend;
        timb_dist(idx) := timb(i);
        idx := idx+1;
    end if;
end loop;

cnt := timb_dist.count;

if cnt = 1 then
    vec.extend();

select *

```

Script Output X | Query Result X

| Task completed in 0.04 seconds

Worksheet Query Builder

```

select *
into vec(1)
from client
where client_id = timb(1);
dbms_output.put_line('Castigatorul este clientul cu id-ul ' || vec(1).client_id);
dbms_output.put_line('Contactati clientul ' || vec(1).client_id || ' la numarul: ' || vec(1).nr_tel_client)
else
    select (trunc(dbms_random.VALUE*cnt)+1)
    into random1
    from dual;

random2 := random1;

while random2 = random1 loop
    select (trunc(dbms_random.VALUE*cnt)+1)
    into random2
    from dual;
end loop;

random1 := timb_dist(random1);
random2 := timb_dist(random2);

vec.extend();
vec.extend();

--adaugam in vector informatiile despre castigatori
select *
into vec(1)
from client
where client_id = random1;

select *
into vec(2)
from client
where client_id = random2;

dbms_output.new_line;
dbms_output.put_line('Cei doi castigatori sunt clientii cu id-urile ' || vec(1).client_id || ' si ' || vec(2).client_id);
dbms_output.put_line('Contactati clientul ' || vec(1).nume_client || ' ' || vec(1).prenume_client || ' cu ' || vec(1).nr_tel_client);
dbms_output.put_line('Contactati clientul ' || vec(2).nume_client || ' ' || vec(2).prenume_client || ' cu ' || vec(2).nr_tel_client);
end if;

```

Script Output X | Query Result X

| Task completed in 0.04 seconds

```

dbms_output.new_line;
dbms_output.put_line('Cei doi castigatori sunt clientii cu id-urile ' || vec(1).client_id || ' si ' || vec(2).client_id);
dbms_output.put_line('Contactati clientul ' || vec(1).nume_client || ' ' || vec(1).prenume_client || ' cu id-ul ' || vec(1).client_id || ' la numarul de telefon: ' || vec(1).nr_tel_client);
dbms_output.put_line('Contactati clientul ' || vec(2).nume_client || ' ' || vec(2).prenume_client || ' cu id-ul ' || vec(2).client_id || ' la numarul de telefon: ' || vec(2).nr_tel_client);

end if;

for i in tidx.first..tidx.last loop
    insert into serviciu
    values tidx(i);
end loop;

else

    dbms_output.put_line('Nu exista candidati la tombola');

end if;
end cerinta_6;
/
begin
cerinta_6;
end;
/

```

Procedure CERINTA_6 compiled

PL/SQL procedure successfully completed.

```

Dbms Output x
+ - Buffer Size: 20000 |
project_an2 x

Clientul 110 candideaza cu vehiculul 120
Clientul 120 candideaza cu vehiculul 130
Clientul 150 candideaza cu vehiculul 170
Clientul 160 candideaza cu vehiculul 180

Cei doi castigatori sunt clientii cu id-urile 120 si 160
Contactati clientul Stoica Isabella cu id-ul 120 la numarul de telefon: 0743123456
Contactati clientul Vasilescu Carol cu id-ul 160 la numarul de telefon: 0787123456

```

Cerinta 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.

Pentru fiecare depozit cu capacitatea strict mai mare de 600 sa se obtina denumirea precum si lista numelor clientilor depozitului impreuna cu denumirea serviciului prestat de catre depozit clientului. Daca depozitul nu are servicii inregistrate sa se precizeze.

```
create or replace procedure cerinta_7 as
--cursor clasic
cursor c is
  select depozit_id, denumire_depozit
  from depozit
  where capacitate_depozit >600;
--ciclu cursor
cursor s (dep_id depozit.depozit_id%type) is
  select s.denumire_serviciu serv, c.nume_client||' '||c.prenume_client nume
  from serviciu s, vehicul v, client c
  where v.vehicul_id=s.vehicul_id and v.client_id = c.client_id
  and s.depozit_id=dep_id
  order by v.vehicul_id;

id_depozit depozit.depozit_id%type;
nume_depozit depozit.denumire_depozit%type;
cnt number:=0;

begin
  open c;
  loop
    fetch c into id_depozit, nume_depozit;
    exit when c%notfound;
    dbms_output.put_line(nume_depozit);
    dbms_output.put_line('-----');
    cnt:=0;
    for j in s(id_depozit) loop
      dbms_output.put_line(j.nume||' - '||j.serv);
      cnt:=cnt+1;
    end loop;

    if cnt =0 then
      dbms_output.put_line('In acest depozit nu s-au prestat servicii');
    end if;
    dbms_output.new_line;
  end loop;
  close c;
end;
/
begin
  cerinta_7;
end;
/
```

Urmeaza printscreen-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul.

 | Task completed in 0.031 seconds

Procedure CERINTA_7 compiled

PL/SQL procedure successfully completed.

Worksheet Query Builder

```

create or replace procedure cerinta_7 as
--cursor clasic
  cursor c is
    select depozit_id, denumire_depozit
    from depozit
    where capacitate_depozit >600;
--ciclu cursor
  cursor s (dep_id depozit.depozit_id%type) is
    select s.denumire_serviciu serv, c.nume_client||' '||c.prenume_client nume
    from serviciu s, vehicul v, client c
    where v.vehicul_id=s.vehicul_id and v.client_id = c.client_id
    and s.depozit_id=dep_id
    order by v.vehicul_id;

  id_depozit depozit.depozit_id%type;
  nume_depozit depozit.denumire_depozit%type;
  cnt number:=0;
begin
  open c;
  loop
    fetch c into id_depozit, nume_depozit;
    exit when c%notfound;
    dbms_output.put_line(nume_depozit);
    dbms_output.put_line('-----');
    cnt:=0;
    for j in s(id_depozit) loop
      dbms_output.put_line(j.nume||' - '||j.serv);
      cnt:=cnt+1;
    end loop;
    if cnt =0 then
      dbms_output.put_line('In acest depozit nu s-au prestat servicii');
    end if;
    dbms_output.new_line;
  end loop;
  close c;
end;
/
begin
  cerinta_7;
end;
/

```

Script Output

Dbms Output x | Buffer Size: 20000 |

project_an2 x

```

Depozitul AutoExpert
-----
Popescu Ema - vopsire
Ionescu Iris - schimb ulei
Stan Maia - geometrie roti

Centrul Logistic AutoMax
-----
Ionescu Iris - depozitare
Stoica Isabella - revizie tehnica
Dumitru Lydia - curatare

Depozitul Integrat AutoPro
-----
In acest depozit nu s-au prestat servicii

Depozitul Avansat AutoTech
-----
Ionescu Iris - reparatie parbriz
Popa Julia - depozitare

Complexul de Depozitare AutoPrime
-----
Vasilescu Carol - depozitare
Mihai Iacob - curatare chimica

Depozitul Specializat AutoExcel
-----
Georgescu Oliver - schimbare placute frana

Centrul de Distributie AutoElite
-----
Andrei Darius - schimb ulei

```

Cerinta 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 exceptii proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

Se da numele unui client și să se returneze numarul de depozite distincte la care s-a reparat mașini. În cazul în care nu există niciun client cu numele introdus ca parametru să se ridice o excepție. În cazul în care sunt mai mulți clienti cu același nume să se ridice o excepție în care să se precizeze să se repeleză funcția cu nume și prenume. Funcția cu nume și prenume va merge doar în cazul în care sunt mai mulți clienti cu același nume.

Pentru a exemplifica exceptiile am facut un insert în plus care va fi sters la final.

```
insert into client
values (220, 'Popa', 'Iulia', '6030910240000', '0747128111');

create or replace function cerinta_8
( nume client.nume_client%type, prenume client.prenume_client%type default '-')
return number is
    cnt_dep number;
    cnt_nume number;
    nu_exista exception;
    mai_multi exception;
    doar_nume exception;
begin
    select count(*)
    into cnt_nume
    from client
    where lower(nume_client) = lower(nume);

    if cnt_nume = 0 then
        raise nu_exista;
    end if;

    if cnt_nume > 1 and prenume <> '-' then
        select count(distinct(s.depozit_id))
        into cnt_dep
        from client c, vehicul v, serviciu s
        where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
        and lower(c.nume_client) = lower(nume) and lower(c.prenume_client) = lower(prenume)
        group by c.client_id;

        dbms_output.put_line( 'Numarul de depozite la care a fost ' || nume || ' ' || prenume || ' este: ');

    else
        if cnt_nume > 1 then
            raise mai_multi;
        end if;
        if cnt_nume = 1 and prenume = '-' then
            select count(distinct(s.depozit_id))
            into cnt_dep
            from client c, vehicul v, serviciu s
            where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
            and lower(c.nume_client) = lower(nume)
            group by c.client_id;

            dbms_output.put_line( 'Numarul de depozite la care a fost ' || nume || ' este: ');

        else
            raise doar_nume;
        end if;
    end if;

    return cnt_dep;

exception
    when nu_exista then
```

```

    raise_application_error(-20000, 'Nu exista clienti cu acest nume');
when mai_multi then
    raise_application_error(-20001, 'Exista mai multi clienti cu acelasi nume, incercati din nou adaugand si
prenumele');
when doar_nume then
    raise_application_error(-20002, 'Trebuie sa introduceti doar numele ca parametru');
end cerinta_8;
/

--exemplu pentru exceptia "Nu exista client cu acest nume"
begin
    dbms_output.put_line(cerinta_8('Ana'));
end;
/

--exemplu pentru exceptia "Exista mai multi clienti cu acelasi nume"
begin
    dbms_output.put_line(cerinta_8('Popa'));
end;
/

--exemplu pentru functia apelata cu nume si prenume
begin
    dbms_output.put_line(cerinta_8('Popa', 'Julia'));
end;
/

--exemplu pentru functia fara exceptii
begin
    dbms_output.put_line(cerinta_8('Ionescu'));
end;
/

--exemplu pentru exceptia "Trebuie sa introduceti doar numele ca parametru"
begin
    dbms_output.put_line(cerinta_8('Ionescu', 'Mihai'));
end;
/

--exemplu in care clientul nu a fost la depozite si se afiseaza 0
begin
    dbms_output.put_line(cerinta_8('Radu'));
end;
/

delete from client
where client_id = 220;

```

Urmeaza printscreens-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul exemplelor.

SQL Worksheet | History

Worksheet | Query Builder

```

insert into client
values (220, 'Popa', 'Iulia', '6030910240000', '0747128111');

create or replace function cerinta_8
  (nume client.nume_client%type, prenume client.prenume_client%type default '-')
  return number is
  cnt_dep number;
  cnt_num number;
  nu_exista exception;
  mai_multi exception;
  doar_nume exception;
begin
  select count(*)
  into cnt_num
  from client
  where lower(nume_client) = lower(nume);

  if cnt_num = 0 then
    raise nu_exista;
  end if;

  if cnt_num > 1 and prenume <> '-' then
    select count(distinct(s.deposit_id))
    into cnt_dep
    from client c, vehicul v, serviciu s
    where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
    and lower(c.nume_client) = lower(nume) and lower(c.prenume_client) = lower(prenume)
    group by c.client_id;

    dbms_output.put_line( 'Numarul de depozite la care a fost ' || nume || ' ' || prenume || ' este: ' );
  else
    if cnt_num > 1 then
      raise mai_multi;
    end if;
    if cnt_num = 1 and prenume = '-' then
      select count(distinct(s.deposit_id))
      into cnt_dep
      from client c, vehicul v, serviciu s
      where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
      and lower(c.nume_client) = lower(nume)
    end if;
  end if;

  return cnt_dep;
exception
  when nu_exista then
    raise_application_error(-20000, 'Nu exista clienti cu acest nume');
  when mai_multi then
    raise_application_error(-20001, 'Există mai multi clienti cu același nume, încercăți din nou adăugând și prenumele');
  when doar_nume then
    raise_application_error(-20002, 'Trebuie să introduceti doar numele ca parametru');
end cerinta_8;
/

```

Worksheet | Query Builder

```

where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
and lower(c.nume_client) = lower(nume)
group by c.client_id;

dbms_output.put_line( 'Numarul de depozite la care a fost ' || nume || ' este: ' );
else
  raise doar_nume;
end if;
end if;

return cnt_dep;

exception
  when nu_exista then
    raise_application_error(-20000, 'Nu există clienti cu acest nume');
  when mai_multi then
    raise_application_error(-20001, 'Există mai multi clienti cu același nume, încercăți din nou adăugând și prenumele');
  when doar_nume then
    raise_application_error(-20002, 'Trebuie să introduceti doar numele ca parametru');
end cerinta_8;
/

--exemplu pentru exceptia "Nu există client cu acest nume"
begin
  dbms_output.put_line(cerinta_8('Ana'));
end;
/

--exemplu pentru exceptia "Există mai multi clienti cu același nume"
begin
  dbms_output.put_line(cerinta_8('Popa'));
end;
/

--exemplu pentru funcția apelată cu nume și prenume
begin
  dbms_output.put_line(cerinta_8('Popa', 'Iulia'));
end;
/

--exemplu pentru funcția fără exceptii

```

```

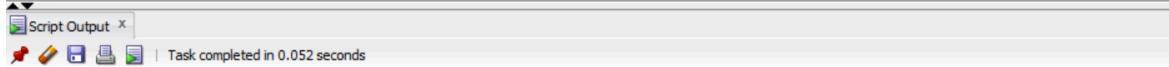
--exemplu pentru funcția fără exceptii
begin
  dbms_output.put_line(cerinta_8('Ionescu'));
end;
/

--exemplu pentru exceptia "Trebuie să introduceti doar numele ca parametru"
begin
  dbms_output.put_line(cerinta_8('Ionescu', 'Mihai'));
end;
/

--exemplu în care clientul nu a făcut la depozite și se afisează 0
begin
  dbms_output.put_line(cerinta_8('Radu'));
end;
/

delete from client
where client_id = 220;

```



```
--exemplu pentru exceptia "Nu exista client cu acest nume"
begin
    dbms_output.put_line(cerinta_8('Ana'));
end;
/

```

Script Output | Query Result | Task completed in 0.029 seconds

```
Error starting at line : 272 in command -
begin
    dbms_output.put_line(cerinta_8('Ana'));
end;
Error report -
ORA-20000: Nu exista clienti cu acest nume
ORA-06512: at "PROIECT2.CERINTA_8", line 47
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
        was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
          the application administrator or DBA for more information.
```

```
--exemplu pentru exceptia "Exista mai multi clienti cu acelasi nume"
begin
    dbms_output.put_line(cerinta_8('Popa'));
end;
/

```

Script Output | Query Result | Task completed in 0.067 seconds

```
Error starting at line : 278 in command -
begin
    dbms_output.put_line(cerinta_8('Popa'));
end;
Error report -
ORA-20001: Exista mai multi clienti cu acelasi nume, incercati din nou adaugand si prenumele
ORA-06512: at "PROIECT2.CERINTA_8", line 49
ORA-06512: at line 2
```

Worksheet | Query Builder | project_an2

```
--exemplu pentru functia apelata cu nume si prenume
begin
    dbms_output.put_line(cerinta_8('Popa', 'Julia'));
end;
/

```

Script Output | Query Result | Task completed in 0.034 seconds

```
PL/SQL procedure successfully completed.
```

project_an2

```
Numarul de depozite la care a fost Popa Julia este:
2
```

Worksheet Query Builder

```
--exemplu pentru functia fara exceptii
begin
    dbms_output.put_line(cerinta_8('Ionescu'));
end;
/
```

Script Output X | Query Result X

PL/SQL procedure successfully completed.

project_an2 x

Numarul de depozite la care a fost Ionescu este:
3

```
--exemplu pentru exceptia "Trebuie sa introduceti doar numele ca parametru"
begin
    dbms_output.put_line(cerinta_8('Ionescu', 'Mihai'));
end;
/
```

Script Output X

Task completed in 0.044 seconds

```
Error starting at line : 302 in command -
begin
    dbms_output.put_line(cerinta_8('Ionescu', 'Mihai'));
end;
Error report -
ORA-20002: Trebuie sa introduceti doar numele ca parametru
ORA-06512: at "PROIECT2.CERINTA_8", line 55
ORA-06512: at line 2
```

Worksheet Query Builder

```
--exemplu in care clientul nu a fost la depozite si se afiseaza 0
begin
    dbms_output.put_line(cerinta_8('Radu'));
end;
/
```

Script Output X | Query Result X

PL/SQL procedure successfully completed.

project_an2 x

Numarul de depozite la care a fost Radu este:
0

La final stergem inserarea facuta la inceput.

```
delete from client
where client_id = 220;
```

Script Output X

Task completed in 0.031 seconds

1 row deleted.

Cerinta 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 exceptiile care pot apărea, inclusiv exceptiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Pentru numele unui proprietar să se afiseze numele tarii și id-ul regiunii unde se află depozitul lui. Sa se trateze urmatoarele exceptii: se poate să nu existe proprietar cu un anumit nume, se poate ca un anumit proprietar existent să nu aibă depozite înregistrate, se poate să existe mai mulți proprietari cu același nume sau se poate ca un proprietar să detină mai multe depozite.

Pentru a exemplifica exceptiile am facut un insert în plus care va fi sters la final.

```
insert into proprietar
values (110, 'Tintas', 'Iulia', 'trebuie sters');

create or replace procedure cerinta_9
( nume proprietar.nume_proprietar%type ) as
  denumire_tara tara.nume_tara%type;
  id_regiune tara.regiune_id%type;
  cnt number := 0;
  id proprietar.proprietar_id%type;
begin
  select count(*)
  into cnt
  from proprietar
  where lower(nume_proprietar) = lower(nume);

  --testare pentru exceptii
  select proprietar_id
  into id
  from proprietar
  where lower(nume_proprietar) = lower(nume);

  select t.nume_tara, t.regiune_id
  into denumire_tara, id_regiune
  from proprietar p, depozit d, locatie l , provincie pr, tara t
  where p.proprietar_id = d.proprietar_id and d.locatie_id = l.locatie_id
  and l.provincie_id = pr.provincie_id and pr.tara_id = t.tara_id
  and lower(p.nume_proprietar) = lower(nume);

  dbms_output.put_line( 'Depozitul proprietarului cu numele'|| nume ||' se află în tara '|| denumire_tara ||' în
  regiunea '|| id_regiune);

exception
  when no_data_found then
    if cnt = 0 then
      raise_application_error(-20003, 'Nu există proprietar cu acest nume');
    else
      raise_application_error(-20004, 'Acest proprietar nu are depozite înregistrate');
    end if;
  when too_many_rows then
    if cnt > 1 then
      raise_application_error(-20005, 'Există mai mulți proprietari cu acest nume');
    else
      raise_application_error(-20006, 'Acest proprietar are mai multe depozite');
    end if;
end;
/
--exemplu în care nu se ridică exceptii
begin
  cerinta_9('albu');
end;
/
```

```

--exemplu pentru excepti "Acest proprietar nu are depozite inregistrate"
begin
    cerinta_9('munteanu');
end;
/

--exemplu pentru excepti "Acest proprietar are mai multe depozite"
begin
    cerinta_9('popescu');
end;
/

--exemplu pentru excepti "Nu exista proprietar cu acest nume"
begin
    cerinta_9('livian');
end;
/

--exemplu pentru excepti "Exista mai multi proprietari cu acest nume"
begin
    cerinta_9('tintas');
end;
/

delete from proprietar
where proprietar_id = 110;

```

Urmeaza printscreens-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul exemplelor.

The screenshot shows the Oracle SQL Developer interface with the 'Worksheet' tab selected. The code area contains the following PL/SQL code:

```

insert into proprietar
values (110, 'Tintas', 'Iulia', 'trebuie sters');

create or replace procedure cerinta_9
    (nume proprietar.nume_proprietar%type)
    as
        denumire_tara tara.nume_tara%type;
        id_regiune tara.regiune_id%type;
        cnt number := 0;
        id proprietar.proprietar_id%type;
    begin
        select count(*)
        into cnt
        from proprietar
        where lower(nume_proprietar) = lower(nume);

        --testare pentru exceptii
        select proprietar_id
        into id
        from proprietar
        where lower(nume_proprietar) = lower(nume);

        select t.nume_tara, t.regiune_id
        into denumire_tara, id_regiune
        from proprietar p, deposit d, locatie l , provincie pr, tara t
        where p.proprietar_id = d.proprietar_id and d.locatie_id = l.locatie_id
        and l.provincie_id = pr.provincie_id and pr.tara_id = t.tara_id
        and lower(p.nume_proprietar) = lower(nume);

        dbms_output.put_line( 'Depozitul proprietarului cu numele '|| nume || ' se afla in tara ' || denumire_tara || ' in regiunea ' || id_regiune);

        exception
            when no_data_found then
                if cnt = 0 then
                    raise_application_error(-20003, 'Nu exista proprietar cu acest nume');
                else
                    raise_application_error(-20004, 'Acest proprietar nu are depozite inregistrate');
                end if;
            when too_many_rows then
                if cnt > 1 then
                    raise_application_error(-20005, 'Exista mai multi proprietari cu acest nume');
                else

```

The results pane below the code shows the output of the procedure execution:

```

Depozitul proprietarului cu numele Tintas se afla in tara Romania in regiunea 1

```

Worksheet | Query Builder

```

        raise_application_error(-20005, 'Exista mai multi proprietari cu acest nume');
    else
        raise_application_error(-20006, 'Acest proprietar are mai multe depozite');
    end if;
end;
/

--exemplu in care nu se ridica exceptii
begin
    cerinta_9('albu');
end;
/

--exemplu pentru exceptii "Acest proprietar nu are depozite inregistrate"
begin
    cerinta_9('munteanu');
end;
/

--exemplu pentru exceptii "Acest proprietar are mai multe depozite"
begin
    cerinta_9('popescu');
end;
/

--exemplu pentru exceptii "Nu exista proprietar cu acest nume"
begin
    cerinta_9('livian');
end;
/

--exemplu pentru exceptii "Exista mai multi proprietari cu acest nume"
begin
    cerinta_9('tintas');
end;
/

delete from proprietar
where proprietar_id = 110;

```

Script Output X | Task completed in 0.038 seconds

Procedure CERINTA_9 compiled

Compiler Log

Worksheet | Query Builder

```

--exemplu in care nu se ridica exceptii
begin
    cerinta_9('albu');
end;
/

```

Script Output X | Query Result X | Task completed in 0.034 seconds

PL/SQL procedure successfully completed.

project_an2 x

Depozitul proprietarului cu numele albu se afla in tara Brazilia in regiunea 5

Worksheet | Query Builder

```

--exemplu pentru exceptii "Acest proprietar nu are depozite inregistrate"
begin
    cerinta_9('munteanu');
end;
/

```

Script Output X | Query Result X | Task completed in 0.051 seconds

PL/SQL procedure successfully completed.

Error starting at line : 364 in command -

```

begin
    cerinta_9('munteanu');
end;
Error report -
ORA-20001: Acest proprietar nu are depozite inregistrate
ORA-06512: at "PROJECT2.CERINTA_9", line 32
ORA-06512: at line 2

```

Worksheet Query Builder

```
--exemplu pentru excepti "Acest proprietar are mai multe depozite"
begin
    cerinta_9('popescu');
end;
/
```

Script Output Query Result Task completed in 0.036 seconds

```
Error starting at line : 370 in command -
begin
    cerinta_9('popescu');
end;
Error report -
ORA-20003: Acest proprietar are mai multe depozite
ORA-06512: at "PROIECT2.CERINTA_9", line 37
ORA-06512: at line 2
```

Worksheet Query Builder

```
--exemplu pentru excepti "Nu exista proprietar cu acest nume"
begin
    cerinta_9('livian');
end;
/
```

Script Output Query Result Task completed in 0.029 seconds

```
Error starting at line : 376 in command -
begin
    cerinta_9('livian');
end;
Error report -
ORA-20000: Nu exista proprietar cu acest nume
ORA-06512: at "PROIECT2.CERINTA_9", line 30
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.
```

Worksheet Query Builder

```
--exemplu pentru excepti "Exista mai multi proprietari cu acest nume"
begin
    cerinta_9('tintas');
end;
/
```

Script Output Query Result Task completed in 0.057 seconds

```
Error starting at line : 382 in command -
begin
    cerinta_9('tintas');
end;
Error report -
ORA-20002: Exista mai multi proprietari cu acest nume
ORA-06512: at "PROIECT2.CERINTA_9", line 35
ORA-06512: at line 2
```

La final stergem inserarea facuta la inceput.

```
delete from proprietar
where proprietar_id = 110;
```

Script Output Task completed in 0.048 seconds

```
1 row deleted.
```

Cerinta 10.

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

Sa nu se permita inserarea in tabelul LUCREAZA intre 24 decembrie si 2 ianuarie.

```
create or replace trigger cerinta_10
  before insert on lucreaza
begin
  if ( (to_number(to_char(sysdate, 'dd')) in (24, 25, 26, 27, 28, 29, 30, 31) and to_number(to_char(sysdate, 'mm')) =
12) or
       (to_number(to_char(sysdate, 'dd')) in (1, 2) and to_number(to_char(sysdate, 'mm')) = 1) ) then
    raise_application_error(-20008, 'Nu se pot face actualizari in tabel in aceasta data');
  end if;
end;
/
insert into lucreaza
values(600,1000,107,'CON', '10-JAN-10',2200);
drop trigger cerinta_10;

drop trigger cerinta_10;
```

Mai sus este este codul care respecta enuntul insa pentru a putea declansa triggerul va trebui sa adaug si data de azi (11 ianuarie) pentru a demonstra functionalitatea lui.

Prin urmare, aici este codul trigger-ului creat doar pentru a putea fi declansat:

```
create or replace trigger cerinta_10
  before insert on lucreaza
begin
  if ( (to_number(to_char(sysdate, 'dd')) in (24, 25, 26, 27, 28, 29, 30, 31) and to_number(to_char(sysdate, 'mm')) =
12) or
       (to_number(to_char(sysdate, 'dd')) in (1, 2, 11) and to_number(to_char(sysdate, 'mm')) = 1) ) then
    raise_application_error(-20008, 'Nu se pot face actualizari in tabel in aceasta data');
  end if;
end;
/
insert into lucreaza
values(600,1000,107,'CON', '10-JAN-10',2200);
drop trigger cerinta_10;

drop trigger cerinta_10;
```

Urmeaza printscreens-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul.

In printscreens se poate observa adaugarea facuta in interiorul instructiunii if. Am facut printscreens-ul astfel incat in coltul din dreapta jos sa se poate observa clar data rularii si anume 11 ianuarie pentru a demonstra corectitudinea trigger-ului.

Oracle SQL Developer : C:\Users\Julia\Desktop\VGBD\exercitii_proiect.sql

File Edit View Navigate Run Source Team Tools Window Help

SQL Worksheet History

Worksheet Query Builder

```
create or replace trigger cerinta_10
    before insert on lucreaza
begin
    if ( to_number(to_char(sysdate, 'dd')) in (24, 25, 26, 27, 28, 29, 30, 31) and to_number(to_char(sysdate, 'mm')) = 12) or
        (to_number(to_char(sysdate, 'dd')) in (1, 2, 11) and to_number(to_char(sysdate, 'mm')) = 1) then
        raise_application_error(-20008, 'Nu se pot face actualizari in tabel in aceasta data');
    end if;
end;
/
insert into lucreaza
values(600,1000,107,'CON', '10-JAN-10',2200);
drop trigger cerinta_10;
```

Script Output

All Results

```
Trigger CERINTA_10 compiled

Error starting at line : 413 in command -
insert into lucreaza
values(600,1000,107,'CON', '10-JAN-10',2200)
Error at Command Line : 413 Column : 13
Error report -
SQL Error: ORA-20008: Nu se pot face actualizari in tabel in aceasta data
ORA-06512: at "PROJECT2.CERINTA_10", line 4
ORA-04088: error during execution of trigger 'PROJECT2.CERINTA_10'

Trigger CERINTA_10 dropped.
```

Search

11:05 AM
1/11/2024

Cerinta 11.

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

Sa nu se permita inserarea vehiculelor mai vechi de 2010 sau a vehiculelor mai noi de 2022 care depasesc 400 de cai putere. Sa nu se permita nici update-uri prin care se maresteste numarul de cai putere cu mai mult de 100 de unitati.

```
create or replace trigger cerinta_11
before insert or update on vehicul
for each row
declare
    cp_old vehicul.cai_putere%type;
    cp_new vehicul.cai_putere%type;
    an_old vehicul.an_fabricatie%type;
    an_new vehicul.an_fabricatie%type;

    expectie_vechime exception;
    exceptie_nou_prea_puternic exception;
    exceptie_diferenta_mare exception;
begin
    cp_old := :OLD.cai_putere;
    cp_new := :NEW.cai_putere;
    an_old := :OLD.an_fabricatie;
    an_new := :NEW.an_fabricatie;
    if inserting then
        if an_new < 2010 then
            raise expectie_vechime;
        end if;
        if an_new > 2022 and cp_new > 400 then
            raise exceptie_nou_prea_puternic;
        end if;
    end if;

    if updating then
        if (cp_new - cp_old > 100) then
            raise exceptie_diferenta_mare;
        end if;
    end if;

exception
    when expectie_vechime then
        raise_application_error (-20009, 'Nu se accepta masini mai vechi de 2010');
    when exceptie_nou_prea_puternic then
        raise_application_error (-20010, 'Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400
de cp');
    when exceptie_diferenta_mare then
        raise_application_error (-20011, 'Nu se permite actualizarea puterii cu mai mult de 100 de cai');
end;
/
--exemplu pentru exceptia 'Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400 de cp'
insert into vehicul
values(1, 100, 'masina', 'orice', 2023, 'Alb', 410, 106);

--exemplu pentru exceptia 'Nu se accepta masini mai vechi de 2010'
insert into vehicul
values(1, 100, 'masina', 'orice', 2000, 'Alb', 100, 106);

--exemplu pentru exceptia 'Nu se permite actualizarea puterii cu mai mult de 100 de cai'
update vehicul
set cai_putere = 500
where vehicul_id = 100;
```

```
drop trigger cerinta_11;
```

Urmeaza printscreen-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul exemplelor.

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a toolbar with various icons. Below it is a tab bar with 'Worksheet' and 'Query Builder'. The main area contains the following PL/SQL code:

```
create or replace trigger cerinta_11
before insert or update on vehicul
for each row
declare
    cp_old vehicul.cai_putere%type;
    cp_new vehicul.cai_putere%type;
    an_old vehicul.an_fabricatie%type;
    an_new vehicul.an_fabricatie%type;

    expectie_vechime exception;
    exceptie_nou_prea_puternic exception;
    exceptie_diferenta_mare exception;
begin
    cp_old := :OLD.cai_putere;
    cp_new := :NEW.cai_putere;
    an_old := :OLD.an_fabricatie;
    an_new := :NEW.an_fabricatie;
    if inserting then
        if an_new < 2010 then
            raise expectie_vechime;
        end if;
        if an_new > 2022 and cp_new > 400 then
            raise exceptie_nou_prea_puternic;
        end if;
    end if;

    if updating then
        if (cp_new - cp_old > 100) then
            raise exceptie_diferenta_mare;
        end if;
    end if;

exception
    when expectie_vechime then
        raise_application_error (-20009, 'Nu se accepta masini mai vechi de 2010');
    when exceptie_nou_prea_puternic then
        raise_application_error (-20010, 'Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400 de cai');
    when exceptie_diferenta_mare then
        raise_application_error (-20011, 'Nu se permite actualizarea puterii cu mai mult de 100 de cai');
end;
/
--exemplu pentru exceptia 'Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400 de cai'
```

The screenshot shows the 'Script Output' tab at the bottom of the interface. It displays the following SQL statements:

```
--exemplu pentru exceptia 'Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400 de cai'
insert into vehicul
values(1, 100, 'masina', 'orice', 2023, 'Alb', 410, 106);

--exemplu pentru exceptia 'Nu se accepta masini mai vechi de 2010'
insert into vehicul
values(1, 100, 'masina', 'orice', 2000, 'Alb', 100, 106);

--exemplu pentru exceptia 'Nu se permite actualizarea puterii cu mai mult de 100 de cai'
update vehicul
set cai_putere = 500
where vehicul_id = 100;

drop trigger cerinta_11;
```

The screenshot shows the 'Script Output' tab at the bottom of the interface. It displays the following SQL statements and an error message:

```
--exemplu pentru exceptia 'Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400 de cai'
insert into vehicul
values(1, 100, 'masina', 'orice', 2023, 'Alb', 410, 106);

Error starting at line : 475 in command -
insert into vehicul
values(1, 100, 'masina', 'orice', 2023, 'Alb', 410, 106)
Error at Command Line : 475 Column : 13
Error report -
SQL Error: ORA-20010: Pentru masini mai noi de 2022 nu se accepta cu mai mult de 400 de cai
ORA-06512: at "PROIECT2.CERINTA_11", line 34
ORA-04088: error during execution of trigger 'PROIECT2.CERINTA_11'
```



Worksheet

Query Builder

```
--exemplu pentru exceptia 'Nu se accepta masini mai vechi de 2010'
insert into vehicul
values(1, 100, 'masina', 'orice', 2000, 'Alb', 100, 106);
```

Script Output X

Task completed in 0.028 seconds

```
Error starting at line : 479 in command -
insert into vehicul
values(1, 100, 'masina', 'orice', 2000, 'Alb', 100, 106)
Error at Command Line : 479 Column : 13
Error report -
SQL Error: ORA-20009: Nu se accepta masini mai vechi de 2010
ORA-06512: at "PROIECT2.CERINTA_11", line 32
ORA-04088: error during execution of trigger 'PROIECT2.CERINTA_11'
```

Worksheet

Query Builder

```
--exemplu pentru exceptia 'Nu se permite actualizarea puterii cu mai mult de 100 de cai'
update vehicul
set cai_putere = 500
where vehicul_id = 100;

drop trigger cerinta_11;
```

Script Output X

Task completed in 0.028 seconds

```
Error starting at line : 483 in command -
update vehicul
set cai_putere = 500
where vehicul_id = 100
Error at Command Line : 483 Column : 8
Error report -
SQL Error: ORA-20011: Nu se permite actualizarea puterii cu mai mult de 100 de cai
ORA-06512: at "PROIECT2.CERINTA_11", line 36
ORA-04088: error during execution of trigger 'PROIECT2.CERINTA_11'
```

Trigger CERINTA_11 dropped.

Se poate observa faptul ca puterea vehiculului cu id-ul 100 este de 120 de cai putere iar update-ul la 500 de cai putere depaseste clar pragul impus de trigger.

```
select *
from vehicul
where vehicul_id = 100;
```

Script Output X

Query Result X

SQL

All Rows Fetched: 1 in 0.063 seconds

VEHICUL_ID	CLIENT_ID	TIP_VEHICUL	MARCA	AN_FABRICATIE	CULOARE	CAI_PUTERE	REPREZENTANTA_ID
1	100	100 masina	Mercedes-Benz	2004 Alb		120	106

Cerinta 12.

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

Creati 3 tabele pentru actiunile de create, drop si alter. Creati un trigger care sa introduca date in aceste tabele in functie de ce comanda LDD a folosit utilizatorul.

```
create table creaza (
    utilizator varchar2(30),
    nume_bd varchar2(50),
    nume_object varchar2(30),
    data date
);

create table sterge (
    utilizator varchar2(30),
    nume_bd varchar2(50),
    nume_object varchar2(30),
    data date
);

create table schimba (
    utilizator varchar2(30),
    nume_bd varchar2(50),
    nume_object varchar2(30),
    data date
);

create or replace trigger cerinta_12
after create or alter or drop on schema
declare

begin
    if sys.sysevent = 'CREATE' then
        insert into creaza
            values (sys.login_user, sys.database_name, sys.dictionary_obj_name, sysdate);
    end if;

    if sys.sysevent = 'DROP' then
        insert into sterge
            values (sys.login_user, sys.database_name, sys.dictionary_obj_name, sysdate);
    end if;

    if sys.sysevent = 'ALTER' then
        insert into schimba
            values (sys.login_user, sys.database_name, sys.dictionary_obj_name, sysdate);
    end if;
end;
/

select *
from creaza;

select *
from sterge;

select *
from schimba;

drop trigger cerinta_12;
```

Urmeaza printscreens-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul.

```

create table creaza (
    utilizator varchar2(30),
    nume_bd varchar2(50),
    nume_object varchar2(30),
    data date
);

create table sterge (
    utilizator varchar2(30),
    nume_bd varchar2(50),
    nume_object varchar2(30),
    data date
);

create table schimba (
    utilizator varchar2(30),
    nume_bd varchar2(50),
    nume_object varchar2(30),
    data date
);

```

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Worksheet" and contains three separate SQL scripts for creating tables: "creaza", "sterge", and "schimba". Each script defines a table with columns for "utilizator" (varchar2(30)), "nume_bd" (varchar2(50)), "nume_object" (varchar2(30)), and "data" (date). The bottom window is titled "Script Output" and shows the message "Task completed in 0.031 seconds".

Table CREAZA created.

Table STERGE created.

Table SCHIMBA created.

```

create or replace trigger cerinta_12
    after create or alter or drop on schema
    declare

begin
    if sys.event = 'CREATE' then
        insert into creaza
        values (sys.login_user, sys.database_name, sys.dictionary_obj_name, sysdate);
    end if;

    if sys.event = 'DROP' then
        insert into sterge
        values (sys.login_user, sys.database_name, sys.dictionary_obj_name, sysdate);
    end if;

    if sys.event = 'ALTER' then
        insert into schimba
        values (sys.login_user, sys.database_name, sys.dictionary_obj_name, sysdate);
    end if;
end;
/

```

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Worksheet" and contains a single SQL script for creating a trigger named "cerinta_12". The trigger is defined as an "OR REPLACE" trigger that fires "after create or alter or drop on schema". It uses the "sys" package to check the event type ("event") and insert records into the corresponding tables ("creaza", "sterge", or "schimba") based on the event type. The bottom window is titled "Script Output" and shows the message "Task completed in 0.031 seconds".

Trigger CERINTA_12 compiled

Pentru a exemplifica functionalitatea declansatorului am sters toate tabelele din proiect si le-am creat iar si am introdus datele necesare. In urma acestor comenzi tabelele creaza, sterge si schimba arata astfel:

```
select *
from creaza;
```

Script Output | Query Result | All Rows Fetched: 47 in 0.001 seconds

UTILIZATOR	NUME_BD	NUME_OBIECT	DATA
1 PROIECT2	XE	REGIUNE	11-JAN-24
2 PROIECT2	XE	REGIUNE_ID_PK	11-JAN-24
3 PROIECT2	XE	TARA_C_ID_PK	11-JAN-24
4 PROIECT2	XE	TARA	11-JAN-24
5 PROIECT2	XE	PROVINCIE_ID_PK	11-JAN-24
6 PROIECT2	XE	PROVINCIE	11-JAN-24
7 PROIECT2	XE	LOCATIE	11-JAN-24
8 PROIECT2	XE	LOC_ID_PK	11-JAN-24
9 PROIECT2	XE	COD_POS_UNQ	11-JAN-24
10 PROIECT2	XE	PROPRIETAR	11-JAN-24
11 PROIECT2	XE	PROPR_ID_PK	11-JAN-24
12 PROIECT2	XE	CNP_CLIENT_UNQ	11-JAN-24
13 PROIECT2	XE	CLIENT	11-JAN-24
14 PROIECT2	XE	CLIENT_ID_PK	11-JAN-24
15 PROIECT2	XE	REPREZENTANTA	11-JAN-24
16 PROIECT2	XE	REPREZENTANTA_ID_PK	11-JAN-24
17 PROIECT2	XE	EMAIL_ANG_UNQ	11-JAN-24
18 PROIECT2	XE	CNP_ANG_UNQ	11-JAN-24
19 PROIECT2	XE	ANGAJAT	11-JAN-24
20 PROIECT2	XE	ANGAJAT_ID_PK	11-JAN-24
21 PROIECT2	XE	JOB	11-JAN-24
22 PROIECT2	XE	JOB_ID_PK	11-JAN-24
23 PROIECT2	XE	VEHICUL	11-JAN-24
24 PROIECT2	XE	VEHICUL_ID_PK	11-JAN-24
25 PROIECT2	XE	VEHIC_TIP_PK	11-JAN-24
26 PROIECT2	XE	VEH_MAS_ID_PK	11-JAN-24
27 PROIECT2	XE	MASINA	11-JAN-24
28 PROIECT2	XE	VEH_BUS_ID_PK	11-JAN-24
29 PROIECT2	XE	AUTOBUZ	11-JAN-24
30 PROIECT2	XE	VEH_TIR_ID_PK	11-JAN-24
31 PROIECT2	XE	TIR	11-JAN-24

```
select *
from sterge;
```

Script Output | Query Result | All Rows Fetched: 25 in 0.004 seconds

UTILIZATOR	NUME_BD	NUME_OBIECT	DATA
1 PROIECT2	XE	SERVICIU	11-JAN-24
2 PROIECT2	XE	SERVICIU_SEQ	11-JAN-24
3 PROIECT2	XE	LUCREAZA	11-JAN-24
4 PROIECT2	XE	LUCREAZA_SEQ	11-JAN-24
5 PROIECT2	XE	DEPOZIT	11-JAN-24
6 PROIECT2	XE	DEPOZIT_SEQ	11-JAN-24
7 PROIECT2	XE	PROPRIETAR	11-JAN-24
8 PROIECT2	XE	PROPRIETAR_SEQ	11-JAN-24
9 PROIECT2	XE	LOCATIE	11-JAN-24
10 PROIECT2	XE	LOCATIE_SEQ	11-JAN-24
11 PROIECT2	XE	PROVINCIE	11-JAN-24
12 PROIECT2	XE	TARA	11-JAN-24
13 PROIECT2	XE	REGIUNE	11-JAN-24
14 PROIECT2	XE	JOB	11-JAN-24
15 PROIECT2	XE	ANGAJAT	11-JAN-24
16 PROIECT2	XE	ANGAJAT_SEQ	11-JAN-24
17 PROIECT2	XE	TIR	11-JAN-24
18 PROIECT2	XE	MASINA	11-JAN-24
19 PROIECT2	XE	AUTOBUZ	11-JAN-24
20 PROIECT2	XE	VEHICUL	11-JAN-24
21 PROIECT2	XE	VEHICUL_SEQ	11-JAN-24
22 PROIECT2	XE	CLIENT	11-JAN-24
23 PROIECT2	XE	CLIENT_SEQ	11-JAN-24
24 PROIECT2	XE	REPREZENTANTA	11-JAN-24
25 PROIECT2	XE	REPREZENTANTA_SEQ	11-JAN-24

```
select *  
from schimba;
```

Script Output X | Query Result X | All Rows Fetched: 17 in 0 seconds

UTILIZATOR	NUME_BD	NUME_OBIECT	DATA
1 PROIECT2	XE	REGIUNE	11-JAN-24
2 PROIECT2	XE	TARA	11-JAN-24
3 PROIECT2	XE	PROVINCIE	11-JAN-24
4 PROIECT2	XE	LOCATIE	11-JAN-24
5 PROIECT2	XE	PROPRIETAR	11-JAN-24
6 PROIECT2	XE	CLIENT	11-JAN-24
7 PROIECT2	XE	REPREZENTANTA	11-JAN-24
8 PROIECT2	XE	ANGAJAT	11-JAN-24
9 PROIECT2	XE	JOB	11-JAN-24
10 PROIECT2	XE	VEHICUL	11-JAN-24
11 PROIECT2	XE	MASINA	11-JAN-24
12 PROIECT2	XE	AUTOBUZ	11-JAN-24
13 PROIECT2	XE	TIR	11-JAN-24
14 PROIECT2	XE	DEPOZIT	11-JAN-24
15 PROIECT2	XE	SERVICIU	11-JAN-24
16 PROIECT2	XE	LUCREAZA	11-JAN-24
17 PROIECT2	XE	DEPOZIT	11-JAN-24

Pentru operatia de stergere a tabelelor am apelat aceste instructiuni:

```
drop table serviciu;  
drop SEQUENCE serviciu_seq;  
drop table lucreaza;  
drop SEQUENCE lucreaza_seq;  
drop table depozit;  
drop SEQUENCE depozit_seq;  
drop table proprietar;  
drop SEQUENCE proprietar_seq;  
drop table locatie;  
drop SEQUENCE locatie_seq;  
drop table provincie;  
drop table tara;  
drop table regiune;  
drop table job;  
drop table angajat;  
drop SEQUENCE angajat_seq;  
drop table tir;  
drop table masina;  
drop table autobuz;  
drop table vehicul;  
drop SEQUENCE vehicul_seq;  
drop table client;  
drop SEQUENCE client_seq;  
drop table reprezentanta;  
drop SEQUENCE reprezentanta_seq;
```

```
drop table serviciu;  
drop SEQUENCE serviciu_seq;  
drop table lucreaza;  
drop SEQUENCE lucreaza_seq;  
drop table depozit;  
drop SEQUENCE depozit_seq;  
drop table proprietar;  
drop SEQUENCE proprietar_seq;  
drop table locatie;  
drop SEQUENCE locatie_seq;  
drop table provincie;  
drop table tara;  
drop table regiune;  
drop table job;  
drop table angajat;  
drop SEQUENCE angajat_seq;  
drop table tir;  
drop table masina;  
drop table autobuz;  
drop table vehicul;  
drop SEQUENCE vehicul_seq;  
drop table client;  
drop SEQUENCE client_seq;  
drop table reprezentanta;  
drop SEQUENCE reprezentanta_seq;
```

Script Output X
| Task completed in 0.361 seconds

Table SERVICIU dropped.

Sequence SERVICIU_SEQ dropped.

Table LUCREAZA dropped.

Sequence LUCREAZA_SEQ dropped.

Table DEPOZIT dropped.

Sequence DEPOZIT_SEQ dropped.

Table PROPRIETAR dropped.

Sequence PROPRIETAR_SEQ dropped.

Table LOCATIE dropped.

Sequence LOCATIE_SEQ dropped.

Table PROVINCIE dropped.

Table TIRI dropped.

Sequence CLIENT_SEQ dropped.

Table REPREZENTANTA dropped.

Sequence REPREZENTANTA_SEQ dropped.

Script Output X
| Task completed in 0.361 seconds

Table TARA dropped.

Table REGIUNE dropped.

Table JOB dropped.

Table ANGAJAT dropped.

Sequence ANGAJAT_SEQ dropped.

Table TIR dropped.

Table MASINA dropped.

Table AUTOBUZ dropped.

Table VEHICUL dropped.

Sequence VEHICUL_SEQ dropped.

Table CLIENT dropped.

Sequence CLIENT_SEQ dropped.

Iar pentru crearea si alterarea tabelelor am apelat instructiunile de la punctul 4 din proiect.

Cerinta 13.

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

```
create or replace package cerinta_13 as
procedure cerinta_6;
procedure cerinta_7;
function cerinta_8 (nume client.nume_client%type, prenume client.prenume_client%type default '-')
    return number;
procedure cerinta_9 (nume proprietar.nume_proprietar%type);
end cerinta_13;
/

create or replace package body cerinta_13 as

procedure cerinta_6 as

type tablou_indexat is table of serviciu%rowtype index by binary_integer;
type tablou_imbricat is table of number;
type vector is varray(20) of client%rowtype;

tidx tablou_indexat;
timb tablou_imbricat := tablou_imbricat();
timb_dist tablou_imbricat := tablou_imbricat();
vec vector := vector();

cnt number;
client_vehicul number;
aux number;
random1 number;
random2 number;
precedent number := 0;
idx number := 1;

begin
--stergere din tabel si salvare in tablou indexat
    delete from serviciu
    where to_char(data_iesire,'yyyy') = '2021'
    returning serviciu_id, depozit_id, vehicul_id, denumire_serviciu, data_introducere, data_iesire
    bulk collect into tidx;

    cnt := tidx.count;
    if cnt <> 0 then
        for i in tidx.first..tidx.last loop
            select client_id
            into client_vehicul
            from vehicul
            where vehicul_id = tidx(i).vehicul_id;
--adaugare id pentru tombola in tabloul imbricat
            timb.extend;
            timb(i) := client_vehicul;
            dbms_output.put_line('Clientul ' || timb(i) || ' candideaza cu vehiculul ' || tidx(i).vehicul_id);
        end loop;

        cnt := timb.count;

--sortam timb
        for i in timb.first..timb.last loop
            for j in (i + 1)..timb.last loop
                if (timb(i) > timb(j)) then
                    aux := timb(i);
                    timb(i) := timb(j);
                    timb(j) := aux;
                end if;
            end loop;
        end loop;

        cnt := timb.count+1;
        idx := 1;
    end if;
end;

```

```

while idx <= timb.last loop
    if timb(idx) = precedent then
        timb.delete(idx);
        cnt := cnt - 1;
        idx := idx +1;
    else
        precedent := timb(idx);
        idx := idx +1;
    end if;
end loop;

cnt := timb.count;
idx := 1;
for i in timb.first..timb.last loop
    if timb.exists(i) then
        timb_dist.extend;
        timb_dist(idx) := timb(i);
        idx := idx+1;
    end if;
end loop;

cnt := timb_dist.count;

if cnt = 1 then
    vec.extend();

    select *
    into vec(1)
    from client
    where client_id = timb(1);
    dbms_output.put_line('Castigatorul este clientul cu id-ul ' || vec(1).client_id);
    dbms_output.put_line('Contactati clientul ' || vec(1).client_id || ' la numarul: ' || vec(1).nr_tel_client);
else
    select (trunc(dbms_random.VALUE*cnt)+1)
    into random1
    from dual;

    random2 := random1;

    while random2 = random1 loop
        select (trunc(dbms_random.VALUE*cnt)+1)
        into random2
        from dual;
    end loop;

    random1 := timb_dist(random1);
    random2 := timb_dist(random2);

    vec.extend();
    vec.extend();

--adaugam in vector informatiile despre castigatori
    select *
    into vec(1)
    from client
    where client_id = random1;

    select *
    into vec(2)
    from client
    where client_id = random2;

    dbms_output.new_line;
    dbms_output.put_line('Cei doi castigatori sunt clientii cu id-urile ' || vec(1).client_id || ' si ' || vec(2).client_id);
    dbms_output.put_line('Contactati clientul ' || vec(1).nume_client || ' ' || vec(1).prenume_client || ' cu id-ul ' || vec(1).client_id || ' la numarul de telefon: ' || vec(1).nr_tel_client);
    dbms_output.put_line('Contactati clientul ' || vec(2).nume_client || ' ' || vec(2).prenume_client || ' cu id-ul ' || vec(2).client_id || ' la numarul de telefon: ' || vec(2).nr_tel_client);
    end if;

```

```

for i in tidx.first..tidx.last loop
    insert into serviciu
        values tidx(i);
end loop;

else
    dbms_output.put_line('Nu exista candidati la tombola');

end if;
end cerinta_6;

procedure cerinta_7 as
--cursor clasic
cursor c is
    select depozit_id, denumire_depozit
    from depozit
    where capacitate_depozit >600;
--ciclu cursor
cursor s (dep_id depozit.depozit_id%type) is
    select s.denumire_serviciu serv, c.nume_client||' '||c.prenume_client nume
    from serviciu s, vehicul v, client c
    where v.vehicul_id=s.vehicul_id and v.client_id = c.client_id
    and s.depozit_id=dep_id
    order by v.vehicul_id;

id_depozit depozit.depozit_id%type;
nume_depozit depozit.denumire_depozit%type;
cnt number:=0;
begin
    open c;
    loop
        fetch c into id_depozit, nume_depozit;
        exit when c%notfound;
        dbms_output.put_line(nume_depozit);
        dbms_output.put_line('-----');
        cnt:=0;
        for j in s(id_depozit) loop
            dbms_output.put_line(j.nume||' - '||j.serv);
            cnt:=cnt+1;
        end loop;

        if cnt =0 then
            dbms_output.put_line('In acest depozit nu s-au prestat servicii');
        end if;
        dbms_output.new_line;
    end loop;
    close c;
end cerinta_7;

function cerinta_8
(num client.nume_client%type, prenume client.prenume_client%type default '-')
return number is
    cnt_dep number;
    cnt_nume number;
    nu_exista exception;
    mai_multi exception;
    doar_nume exception;
begin
    select count(*)
    into cnt_nume
    from client
    where lower(nume_client) = lower(num);

    if cnt_nume =0 then
        raise nu_exista;
    end if;

    if cnt_nume > 1 and prenume <> '-' then
        select count(distinct(s.depozit_id))
        into cnt_dep
    end if;

```

```

from client c, vehicul v, serviciu s
where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
and lower(c.nume_client) = lower(nume) and lower(c.prenume_client) = lower(prenume)
group by c.client_id;

dbms_output.put_line( 'Numarul de depozite la care a fost ' || nume || ' ' || prenume || ' este: ');

else
  if cnt_nume > 1 then
    raise mai_multi;
  end if;
  if cnt_nume = 1 and prenume = '-' then
    select count(distinct(s.depozit_id))
    into cnt_dep
    from client c, vehicul v, serviciu s
    where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
    and lower(c.nume_client) = lower(nume)
    group by c.client_id;

    dbms_output.put_line( 'Numarul de depozite la care a fost ' || nume || ' este: ');
  else
    raise doar_nume;
  end if;
end if;

return cnt_dep;

exception
when nu_exista then
  raise_application_error(-20000, 'Nu exista clienti cu acest nume');
when mai_multi then
  raise_application_error(-20001, 'Există mai multi clienti cu același nume, încercăți din nou adăugând și prenumele');
when doar_nume then
  raise_application_error(-20002, 'Trebuie să introduceti doar numele ca parametru');
end cerinta_8;

procedure cerinta_9
(num proprietar.nume_proprietar%type) as
denumire_tara tara.nume_tara%type;
id_regiune tara.regiune_id%type;
cnt number := 0;
id proprietar.proprietar_id%type;
begin
  select count(*)
  into cnt
  from proprietar
  where lower(nume_proprietar) = lower(nume);

  --testare pentru exceptii
  select proprietar_id
  into id
  from proprietar
  where lower(nume_proprietar) = lower(nume);

  select t.nume_tara, t.regiune_id
  into denumire_tara, id_regiune
  from proprietar p, depozit d, locatie l, provincie pr, tara t
  where p.proprietar_id = d.proprietar_id and d.locatie_id = l.locatie_id
  and l.provincie_id = pr.provincie_id and pr.tara_id = t.tara_id
  and lower(p.nume_proprietar) = lower(nume);

  dbms_output.put_line( 'Depozitul proprietarului cu numele '|| nume || ' se află în țara '|| denumire_tara || ' în
regiunea '|| id_regiune);

exception
when no_data_found then
  if cnt = 0 then
    raise_application_error(-20003, 'Nu există proprietar cu acest nume');
  else
    raise_application_error(-20004, 'Acest proprietar nu are depozite înregistrate');
  end if;
end;

```

```

    end if;
when too_many_rows then
    if cnt > 1 then
        raise_application_error(-20005, 'Exista mai multi proprietari cu acest nume');
    else
        raise_application_error(-20006, 'Acest proprietar are mai multe depozite');
    end if;
end cerinta_9;

end cerinta_13;
/

begin
    dbms_output.put_line('**CERINTA 6**');
    dbms_output.new_line;
    cerinta_13.cerinta_6;
    dbms_output.new_line;
    dbms_output.put_line('-----');
    dbms_output.new_line;
    dbms_output.put_line('**CERINTA 7**');
    dbms_output.new_line;
    cerinta_13.cerinta_7;
    dbms_output.new_line;
    dbms_output.put_line('-----');
    dbms_output.new_line;
    dbms_output.put_line('**CERINTA 8**');
    dbms_output.new_line;
    dbms_output.put_line(cerinta_13.cerinta_8('Ionescu'));
    dbms_output.new_line;
    dbms_output.put_line('-----');
    dbms_output.new_line;
    dbms_output.put_line('**CERINTA 9**');
    dbms_output.new_line;
    cerinta_13.cerinta_9('albu');
    dbms_output.new_line;
    dbms_output.put_line('-----');
end;
/
drop package cerinta_13;

```

Urmeaza printscreen-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul.

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

- Script Editor:** Contains the PL/SQL code for creating the package cerinta_13. The code includes procedures cerinta_6, cerinta_7, and cerinta_9, and a function cerinta_8 that returns a number. The package ends with a slash (/).
- Script Output:** A tab showing the results of the execution. It displays the message "Task completed in 0.028 seconds".
- Bottom Status Bar:** Shows the message "Package CERINTA_13 compiled".

Worksheet | Query Builder

```

create or replace package body cerinta_13 as
procedure cerinta_6 as
begin
--stergere din tabel si salvare in tablou indexat
    delete from serviciu
    where to_char(data_iesire,'yyyy') = '2021'
    returning serviciu_id, depozit_id, vehicul_id, denumire_serviciu, data_introducere, data_iesire
    bulk collect into tidx;
    cnt := tidx.count;
    if cnt <> 0 then
        for i in tidx.first..tidx.last loop
            select client_id
            into client_vehicul
            from vehicul
            where vehicul_id = tidx(i).vehicul_id;
--adaugare id pentru tombola in tabloul imbricat
            timb.extend;
            timb(i) := client_vehicul;
            dbms_output.put_line('Clientul ' || timb(i) || ' candideaza cu vehiculul ' || tidx(i).vehicul_id);
        end loop;
        cnt := timb.count;
    end if;
end;

```

Script Output X

Worksheet | Query Builder

```

end loop;

cnt := timb.count;

--sortam timb
for i in timb.first..timb.last loop
    for j in (i + 1)..timb.last loop
        if (timb(i) > timb(j)) then
            aux := timb(i);
            timb(i) := timb(j);
            timb(j) := aux;
        end if;
    end loop;
end loop;

cnt := timb.count+1;
idx := 1;

while idx <= timb.last loop
    if timb(idx) = precedent then
        timb.delete(idx);
        cnt := cnt - 1;
        idx := idx +1;
    else
        precedent := timb(idx);
        idx := idx +1;
    end if;
end loop;

cnt := timb.count;
idx := 1;
for i in timb.first..timb.last loop
    if timb.exists(i) then
        timb_dist.extend;
        timb_dist(idx) := timb(i);
        idx := idx+1;
    end if;
end loop;

cnt := timb_dist.count;

if cnt = 1 then

```

Script Output X

| Task completed in 0.03 seconds

Worksheet | Query Builder

```

if cnt = 1 then
    vec.extend();

    select *
    into vec(1)
    from client
    where client_id = timb(1);
    dbms_output.put_line('Castigatorul este clientul cu id-ul ' || vec(1).client_id);
    dbms_output.put_line('Contactati clientul ' || vec(1).client_id || ' la numarul: ' || vec(1).nr_tel_client);

else
    select (trunc(dbms_random.VALUE*cnt)+1)
    into random1
    from dual;

    random2 := random1;

    while random2 = random1 loop
        select (trunc(dbms_random.VALUE*cnt)+1)
        into random2
        from dual;
    end loop;

    random1 := timb_dist(random1);
    random2 := timb_dist(random2);

    vec.extend();
    vec.extend();

--adaugam in vector informatiile despre castigator
select *
into vec(1)
from client
where client_id = random1;

select *
into vec(2)
from client
where client_id = random2;

dbms_output.new_line;
dbms_output.put_line('Cei doi castigatori sunt clientii cu id-urile ' || vec(1).client_id || ' si ' || vec(2).client_id);
dbms_output.put_line('Contactati clientul ' || vec(1).nume_client || ' ' || vec(1).prenume_client || ' cu id-ul ' || vec(1).cl

```

Script Output X | Task completed in 0.03 seconds

SQL Worksheet | History

Worksheet | Query Builder

```

dbms_output.put_line('Contactati clientul ' || vec(1).nume_client || ' ' || vec(1).prenume_client || ' cu id-ul ' || vec(1).cl
dbms_output.put_line('Contactati clientul ' || vec(2).nume_client || ' ' || vec(2).prenume_client || ' cu id-ul ' || vec(2).cl
end if;

for i in tidx.first..tidx.last loop
    insert into serviciu
    values tidx(i);
end loop;

else

    dbms_output.put_line('Nu exista candidati la tombola');

end if;
end cerinta_6;

procedure cerinta_7 as
--cursor clasic
cursor c is
    select deposit_id, denumire_depozit
    from depozit
    where capacitate_depozit >600;
--ciclu cursor
cursor s (dep_id depozit.deposit_id%type) is
    select s.denumire_serviciu serv, c.nume_client||' '||c.prenume_client nume
    from serviciu s, vehicul v, client c
    where v.vehicul_id=s.vehicul_id and v.client_id = c.client_id
    and s.deposit_id=dep_id
    order by v.vehicul_id;

id_depozit depozit.deposit_id%type;
nume_depozit depozit.denumire_depozit%type;
cnt number:=0;
begin
    open c;
    loop
        fetch c into id_depozit, nume_depozit;
        exit when c%notfound;
        dbms_output.put_line(nume_depozit);
        dbms_output.put_line('-----');
        cnt:=0;
        for j in s(id_depozit) loop

```

Script Output X | Task completed in 0.03 seconds

SQL Worksheet History

Worksheet Query Builder

```

        for j in s1(q_depozit) loop
            dbms_output.put_line(j.nume||' - ||j.serv);
            cnt:=cnt+1;
        end loop;

        if cnt =0 then
            dbms_output.put_line('In acest depozit nu s-au prestat servicii');
        end if;
        dbms_output.new_line;
    end loop;
    close c;
end cerinta_7;

function cerinta_8
(num client.nume_client%type, prenume client.prenume_client%type default '-')
return number is
    cnt_dep number;
    cnt_nume number;
    nu_exista exception;
    mai_multi exception;
    doar_nume exception;
begin
    select count(*)
    into cnt_nume
    from client
    where lower(nume_client) = lower(num);

    if cnt_nume =0 then
        raise nu_exista;
    end if;

    if cnt_nume > 1 and prenume <> '-' then
        select count(distinct(s.depozit_id))
        into cnt_dep
        from client c, vehicul v, serviciu s
        where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
        and lower(c.nume_client) = lower(num) and lower(c.prenume_client) = lower(prenume)
        group by c.client_id;

        dbms_output.put_line( 'Numarul de depozite la care a fost ' || num || ' ' || prenume || ' este: ');

    else

```

Script Output x | Task completed in 0.03 seconds

Worksheet Query Builder

```

        else
            if cnt_nume > 1 then
                raise mai_multi;
            end if;
            if cnt_nume = 1 and prenume = '-' then
                select count(distinct(s.depozit_id))
                into cnt_dep
                from client c, vehicul v, serviciu s
                where c.client_id = v.client_id(+) and v.vehicul_id = s.vehicul_id(+)
                and lower(c.nume_client) = lower(num)
                group by c.client_id;

                dbms_output.put_line( 'Numarul de depozite la care a fost ' || num || ' este: ');

            else
                raise doar_nume;
            end if;
        end if;

        return cnt_dep;
    exception
        when nu_exista then
            raise_application_error(-20000, 'Nu exista clienti cu acest nume');
        when mai_multi then
            raise_application_error(-20001, 'Exista mai multi clienti cu acelasi nume, incercati din nou adaugand si prenumele');
        when doar_nume then
            raise_application_error(-20002, 'Trebuie sa introduceti doar numele ca parametru');
    end cerinta_8;

procedure cerinta_9
(num proprietar.nume_proprietar%type) as
denumire_tara tara.nume_tara%type;
id_regiune tara.regiune_id%type;
cnt number := 0;
id_proprietar.proprietar_id%type;
begin
    select count(*)
    into cnt
    from proprietar
    where lower(nume_proprietar) = lower(num);

```

Script Output x | Task completed in 0.03 seconds

SQL Worksheet History

Worksheet Query Builder

```
    into cnt
    from proprietar
    where lower(nume_proprietar) = lower(nume);

    --testare pentru exceptii
    select proprietar_id
    into id
    from proprietar
    where lower(nume_proprietar) = lower(nume);

    select t.nume_tara, t.regiune_id
    into denumire_tara, id_regiune
    from proprietar p, deposit d, locatie l, provincie pr, tara t
    where p.proprietar_id = d.proprietar_id and d.locatie_id = l.locatie_id
    and l.provincie_id = pr.provincie_id and pr.tara_id = t.tara_id
    and lower(p.num_proprietar) = lower(nume);

    dbms_output.put_line( 'Depozitul proprietarului cu numele ' || nume || ' se afla in tara ' || denumire_tara || ' in regiunea ' || id_regiune);

    exception
        when no_data_found then
            if cnt = 0 then
                raise_application_error(-20003, 'Nu exista proprietar cu acest nume');
            else
                raise_application_error(-20004, 'Acest proprietar nu are depozite inregistrate');
            end if;
        when too_many_rows then
            if cnt > 1 then
                raise_application_error(-20005, 'Există mai multi proprietari cu acest nume');
            else
                raise_application_error(-20006, 'Acest proprietar are mai multe depozite');
            end if;
    end cerinta_9;

end cerinta_13;
/
```

Script Output X | Task completed in 0.03 seconds

```
Package CERINTA_13 compiled
```

Package Body CERINTA_13 compiled

```
begin
    dbms_output.put_line('***CERINTA 6***');
    dbms_output.new_line;
    cerinta_13.cerinta_6;
    dbms_output.new_line;
    dbms_output.put_line('-----');
    dbms_output.new_line;
    dbms_output.put_line('***CERINTA 7***');
    dbms_output.new_line;
    cerinta_13.cerinta_7;
    dbms_output.new_line;
    dbms_output.put_line('-----');
    dbms_output.new_line;
    dbms_output.put_line('***CERINTA 8***');
    dbms_output.new_line;
    dbms_output.put_line(cerinta_13.cerinta_8('Ionescu'));
    dbms_output.new_line;
    dbms_output.put_line('-----');
    dbms_output.new_line;
    dbms_output.put_line('***CERINTA 9***');
    dbms_output.new_line;
    cerinta_13.cerinta_9('albu');
    dbms_output.new_line;
    dbms_output.put_line('-----');
```

Script Output X | Task completed in 0.032 seconds

```
Package Body CERINTA_13 compiled
```

PL/SQL procedure successfully completed.

Dbms Output | Buffer Size: 20000

proiect_an2

CERINTA 6

Clientul 110 candideaza cu vehiculul 120
Clientul 120 candideaza cu vehiculul 130
Clientul 150 candideaza cu vehiculul 170
Clientul 160 candideaza cu vehiculul 180

Cei doi castigatori sunt clientii cu id-urile 120 si 160
Contactati clientul Stoica Isabella cu id-ul 120 la numarul de telefon: 0743123456
Contactati clientul Vasilescu Carol cu id-ul 160 la numarul de telefon: 0787123456

CERINTA 7

Depozitul AutoExpert

Popescu Ema - vopsire
Ionescu Iris - schimb ulei
Stan Maia - geometrie roti

Centralul Logistic AutoMax

Ionescu Iris - depozitare
Stoica Isabella - revizie tehnica
Dumitru Lydia - curatare

Depozitul Integrat AutoPro

In acest depozit nu s-au prestat servicii

Depozitul Avansat AutoTech

Ionescu Iris - reparatie parbriz
Popa Julia - depozitare

Complexul de Depozitare AutoPrime

Vasilescu Carol - depozitare
Mihai Iacob - curatare chimica

Depozitul Specializat AutoExcel

Georgescu Oliver - schimbare placute frana

Centrul de Distributie AutoElite

Andrei Darius - schimb ulei

Andrei Darius - schimb ulei

CERINTA 8

Numarul de depozite la care a fost Ionescu este:
3

CERINTA 9

Depozitul proprietarului cu numele albu se afla in tara Brazilia in regiunea 5

Cerinta 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).

Am definit un pachet care contine doua tipuri de date de tip record cu ajutorul cărora vom stoca detalii despre un angajat sau un vehicul, două funcții de extras detalii despre angajat și vehicul și două proceduri prin care se fac update-uri angajatului sau vehiculului cu id-ul din parametru.

```
create or replace package cerinta_14 as
type detalii_angajat is record(
    id_angajat.id%type,
    nume_angajat.nume_ang%type,
    prenume_angajat.prenume_ang%type,
    varsta number,
    salar_cumulat number
);
type detalii_vehicul is record(
    id_vehicul.id%type,
    client_vehicul.client_id%type,
    marca_vehicul.marca%type,
    putere_vehicul.cai_putere%type,
    zile_de_stat_in_depozit_cumulat number
);
function get_detalii_angajat ( id number ) return detalii_angajat;
function get_detalii_vehicul ( id number ) return detalii_vehicul;
procedure set_detalii_angajat (id number , nr_tel_angajat.nr_tel_ang%type , varsta_angajat.varsta_ang%type default 0);
procedure set_detalii_vehicul (id number , c_vehicul.culoare%type , putere_vehicul.cai_putere%type default 0);
end cerinta_14;
/
create or replace package body cerinta_14 as
function get_detalii_angajat ( id number ) return detalii_angajat is
    detalii_detalii_angajat;
    nume_angajat.nume_ang%type;
    prenume_angajat.prenume_ang%type;
    varsta number;
    salar number;
begin
    select nume_ang, prenume_ang, varsta
    into nume, prenume, varsta
    from angajat
    where angajat_id = id;

    select sum(salar)
    into salar
    from lucreaza
    where angajat_id = id;

    detalii.id := id;
    detalii.nume := nume;
    detalii.prenume := prenume;
    detalii.varsta := varsta;
    detalii.salar_cumulat := salar;
    return detalii;
exception
    when no_data_found then
        raise_application_error(-20000, 'Nu există angajat cu acest id');
end get_detalii_angajat;
function get_detalii_vehicul ( id number ) return detalii_vehicul is
```

```

detalii detalii_vehicul;
client vehicul.client_id%type;
marca vehicul.marca%type;
putere vehicul.cai_putere%type;
zile_depozit number;
cnt number := 0;
begin
    select client_id, marca, cai_putere
    into client, marca, putere
    from vehicul
    where vehicul_id = id;

    select count(*)
    into cnt
    from serviciu
    where vehicul_id = id;

    if cnt <> 0 then
        select sum(data_iesire - data_introducere)
        into zile_depozit
        from serviciu
        where vehicul_id = id;
    else
        zile_depozit := 0;
    end if;

    detalii.id := id;
    detalii.client := client;
    detalii.marca := marca;
    detalii.putere := putere;
    detalii.zile_de_stat_in_depozit_cumulat := zile_depozit;

    return detalii;
exception
    when no_data_found then
        raise_application_error(-20002, 'Nu exista vehicul cu acest id');
end get_detalii_vehicul;

procedure set_detalii_angajat (id number , nr_tel_angajat.nr_tel_ang%type , varsta_angajat.varsta_ang%type default 0) is
    cnt number;
    nu_este exception;
begin
    select count(*)
    into cnt
    from angajat
    where angajat_id = id;

    if cnt = 0 then
        raise nu_este;
    end if;

    update angajat
    set nr_tel_ang = nr_tel
    where angajat_id = id;

    if varsta <> 0 then
        update angajat
        set varsta_ang = varsta
        where angajat_id = id;
    end if;
exception
    when nu_este then
        raise_application_error(-20003, 'Nu exista angajat cu acest id');
end set_detalii_angajat;

procedure set_detalii_vehicul (id number , c_vehicul.culoare%type , putere_vehicul.cai_putere%type default 0) is
    cnt number;
    nu_este exception;
begin
    select count(*)

```

```

into cnt
from vehicul
where vehicul_id = id;

if cnt = 0 then
    raise nu_este;
end if;

update vehicul
set culoare = c
where vehicul_id = id;

if putere <> 0 then
    update vehicul
    set cai_putere = putere
    where vehicul_id = id;
end if;

exception
when nu_este then
    raise_application_error(-20001, 'Nu exista vehicul cu acest id');
end set_detalii_vehicul;

end cerinta_14;
/

begin
dbms_output.put_line('Salariul cumulat al angajatului cu id-ul 100 este:');
dbms_output.put_line(cerinta_14.get_detalii_angajat(100).salar_cumulat);
dbms_output.new_line;
dbms_output.put_line('Numarul de zile in care vehiculul cu id-ul 110 a stat in depozite este:');
dbms_output.put_line(cerinta_14.get_detalii_vehicul(110).zile_de_stat_in_depozit_cumulat);
dbms_output.new_line;
dbms_output.put_line('Numarul de zile in care vehiculul cu id-ul 230 a stat in depozite este:');
dbms_output.put_line(cerinta_14.get_detalii_vehicul(230).zile_de_stat_in_depozit_cumulat);

cerinta_14.set_detalii_angajat(100, '1234567891');
cerinta_14.set_detalii_angajat(110, '1234', 20);
cerinta_14.set_detalii_vehicul(110, 'curcubeu');
cerinta_14.set_detalii_vehicul(120, 'roz', 250);
end;
/
drop package cerinta_14;

```

Urmeaza printscreens-uri cu codul care sa demonstreze ca acesta ruleaza fara erori impreuna cu outputul.

```

--ex 14

create or replace package cerinta_14 as
    type detalii_angajat is record(
        id_angajat angajat_id%type,
        nume_angajat nume_ang%type,
        prenume_angajat prenume_ang%type,
        varsta number,
        salar_cumulat number
    );

    type detalii_vehicul is record(
        id_vehicul vehicul_id%type,
        client_vehicul client_id%type,
        marca_vehicul marca%type,
        putere_vehicul cai_putere%type,
        zile_de_stat_in_depozit Cumulat number
    );
    
    function get_detalii_angajat ( id number ) return detalii_angajat;
    function get_detalii_vehicul ( id number ) return detalii_vehicul;
    procedure set_detalii_angajat (id number , nr_tel_angajat.nr_tel_ang%type , varsta_angajat.varsta_ang%type default 0);
    procedure set_detalii_vehicul (id number , c_vehicul.culoare%type , putere_vehicul.cai_putere%type default 0);

end cerinta_14;
/

```

Script Output X | Task completed in 0.059 seconds

Package CERINTA_14 compiled

```

create or replace package body cerinta_14 as

    function get_detalii_angajat ( id number ) return detalii_angajat is ...
    function get_detalii_vehicul ( id number ) return detalii_vehicul is ...

procedure set_detalii_angajat (id number , nr_tel_angajat.nr_tel_angajat type , varsta_angajat.varsta_angajat type default 0) is
    cnt number;
    nu_este exception;
begin
    select count(*)
    into cnt
    from angajat
    where angajat_id = id;

    if cnt = 0 then
        raise nu_este;
    end if;

    update angajat
    set nr_tel_ang = nr_tel
    where angajat_id = id;

    if varsta <> 0 then
        update angajat
        set varsta_ang = varsta
        where angajat_id = id;
    end if;
exception
    when nu_este then
        raise_application_error(-20003, 'Nu exista angajat cu acest id');
end set_detalii_angajat;

procedure set_detalii_vehicul (id number , c vehicul.culoare type , putere_vehicul.cai_putere type default 0) is
    cnt number;
    nu_este exception;
begin
    select count(*)
    into cnt
    from vehicul
    where vehicul_id = id;

```

Worksheet | Query Builder

```

        set cai_putere = putere
        where vehicul_id = id;
    end if;
exception
    when nu_este then
        raise_application_error(-20001, 'Nu exista vehicul cu acest id');
    end set_detalii_vehicul;

end cerinta_14;
/

begin
    dbms_output.put_line('Salariul cumulat al angajatului cu id-ul 100 este:');
    dbms_output.put_line(cerinta_14.get_detalii_angajat(100).salar_cumulat);
    dbms_output.new_line;
    dbms_output.put_line('Numarul de zile in care vehiculul cu id-ul 110 a stat in depozite este:');
    dbms_output.put_line(cerinta_14.get_detalii_vehicul(110).zile_de_stat_in_depozit_cumulat);
    dbms_output.new_line;
    dbms_output.put_line('Numarul de zile in care vehiculul cu id-ul 230 a stat in depozite este:');
    dbms_output.put_line(cerinta_14.get_detalii_vehicul(230).zile_de_stat_in_depozit_cumulat);

    cerinta_14.set_detalii_angajat(100, '1234567891');
    cerinta_14.set_detalii_angajat(110, '1234', 20);
    cerinta_14.set_detalii_vehicul(110, 'curcubeu');
    cerinta_14.set_detalii_vehicul(120, 'roz', 250);
end;
/

```

Script Output | Task completed in 0.07 seconds

```

Package CERINTA_14 compiled

Package Body CERINTA_14 compiled

PL/SQL procedure successfully completed.

```

Dbms Output | Buffer Size: 20000

project_an2 *

```

Salariul cumulat al angajatului cu id-ul 100 este:
7010

Numarul de zile in care vehiculul cu id-ul 110 a stat in depozite este:
21

Numarul de zile in care vehiculul cu id-ul 230 a stat in depozite este:
0

```

Pentru a demonstra corectitudinea functiilor si faptul ca update-urile din proceduri au fost facute corect am aceste selecturi:

```

select sum(salar)
from lucreaza
where angajat_id = 100;

```

	SUM(SALAR)
1	7010


```

select sum(data_iesire - data_introducere)
from serviciu
where vehicul_id = 110;

```

	SUM(DATAIESIRE-DATAINTRODUCERE)
1	21


```

select sum(data_iesire - data_introducere)
from serviciu
where vehicul_id = 230;

```

	SUM(DATAIESIRE-DATAINTRODUCERE)
1	(null)

Select-uri realizate inainte de apelarea blocului pl/sql:

```

select *
from vehicul
where vehicul_id in (110, 120);

select *
from angajat
where angajat_id in (100, 110);

```

VEHICUL_ID	CLIENT_ID	TIP_VEHICUL	MARCA	AN_FABRICATIE	CULOARE	CAI_PUTERE	REPREZENTANTA_ID
1	110	110 autobuz	Volvo	2008	Negru	150	103
2	120	110 masina	Volkswagen	2001	Argintiu	180	102

ANGAJAT_ID	NUME_ANG	PRENUME_ANG	NR_TEL_ANG	EMAIL_ANG	CNP_ANG	VARSTA_ANG	
1	100	Dragos	Ana	0722345678	ana.dragos@gmail.com	1890223456789	20
2	110	Preda	Mihai	0822345678	mihai.preda@gmail.com	1991223456789	37

Select-uri realizate dupa apelarea blocului pl/sql:

```

select *
from vehicul
where vehicul_id in (110, 120);

select *
from angajat
where angajat_id in (100, 110);

```

VEHICUL_ID	CLIENT_ID	TIP_VEHICUL	MARCA	AN_FABRICATIE	CULOARE	CAI_PUTERE	REPREZENTANTA_ID
1	110	110 autobuz	Volvo	2008	curcubeu	150	103
2	120	110 masina	Volkswagen	2001	roz	250	102

ANGAJAT_ID	NUME_ANG	PRENUME_ANG	NR_TEL_ANG	EMAIL_ANG	CNP_ANG	VARSTA_ANG	
1	100	Dragos	Ana	1234567891	ana.dragos@gmail.com	1890223456789	20
2	110	Preda	Mihai	1234	mihai.preda@gmail.com	1991223456789	20