**Gestiunea service-urilor auto din Romania**

Nume: Răducanu Andrei-Cosmin

Grupa: 242

**Utilitatea bazei de date**

Exemplele din acest capitol se referă la proiectarea unui model de date ce furnizează informații despre service-urile auto din România, un loc ce este tot mai des frecventat, deoarece numărul mașinilor din România este în continuă creștere, iar acestea trebuie întreținute și reparate constant. Pentru acest model de date voi construi diagrama E/R corespunzătoare, dar și diagrama conceptuală. Modelul de date va gestiona informații legate de organizarea și funcționarea service-urilor auto din România.

Totodată, baza de date este in FN3 si nu prezinta redundanță sau neclarități ce pot duce la alterarea datelor stocate.

Scurtă introducere în modelul de date și restricțiile pe care acesta le respectă pentru funcționare:

Modelul de date va gestiona informații legate de organizarea și funcționarea service-urilor auto din România. Toate acestea au furnizori - ei se află în colaborare cu service-uri pentru a le aproviziona cu piese auto atunci când este nevoie (în baza unui contract). Un anumit tip de piesă auto poate fi furnizată de un singur furnizor.

Fiecare service se află într-o anumită locație și are unul sau mai mulți angajați. Un service nu se poate afla decât într-o locație și numai una.

Angajații au fiecare câte un job și numai unul.

Clienții își fac o programare și vin cu o anumită mașină. Fiecare mașină care vine în service are nevoie de minim o piesă, care este montată de unul sau de mai mulți angajați.

Modelul de date respectă anumite restricții de funcționare:

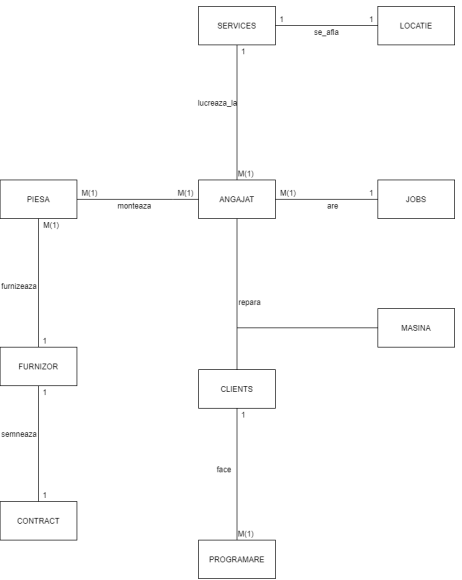
➢ Un angajat poate monta mai multe piese și minim una, iar o piesă trebuie să fie montată de minim un angajat.

➢ Funizorii aprovizionează service-ul auto cu piese auto, vopsea și diferite materiale pe durata contractului.

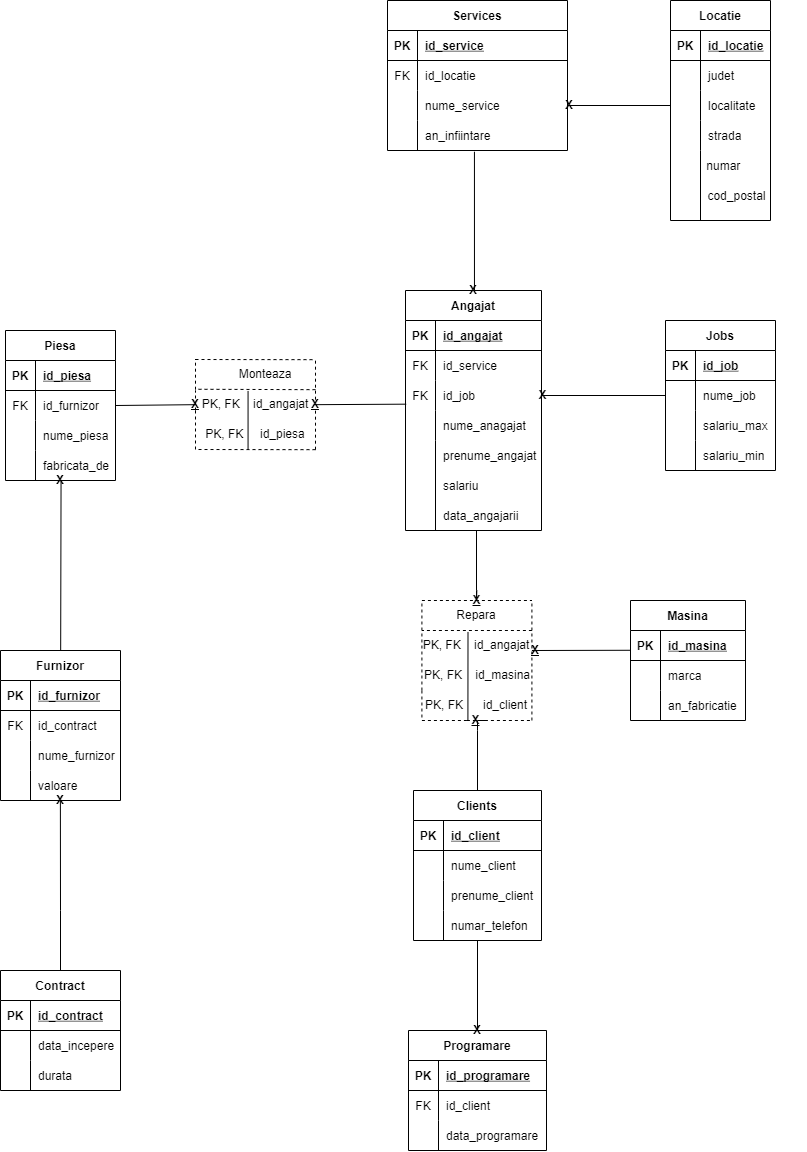
➢ Fiecare angajat trebuie să aibă un job și numai unul.

➢ O piesă este furnizată de un singur furnizor, care poate furniza mai mult de o piesă.

**Diagrama E/R (Entitate - Relație)**



**Diagrama conceptuala (cu toate atributele)**

****

**Exercitiile 4 si 5.**

CREATE TABLE LOCATIE

(id\_locatie number(6),

judet varchar2(20) constraint judet\_nn not null,

localitate varchar2(30) constraint localitate\_nn not null,

strada varchar2(30) constraint strada\_nn not null,

numar number(3) constraint numar\_nn not null,

cod\_postal varchar2(6),

constraint pk\_locatie primary key(id\_locatie),

check(numar > 0)

);

CREATE SEQUENCE SEQUENCE\_LOCATIE

INCREMENT BY 1

START WITH 0

MINVALUE 0

MAXVALUE 100

NOCYCLE;

INSERT INTO LOCATIE VALUES(SEQUENCE\_LOCATIE.NEXTVAL, 'sector 4', 'Bucuresti', 'Giurgiului', 109, '050355');

INSERT INTO LOCATIE VALUES(SEQUENCE\_LOCATIE.NEXTVAL, 'sector 5', 'Bucuresti', 'Ogradei', 4, '050325');

INSERT INTO LOCATIE VALUES(SEQUENCE\_LOCATIE.NEXTVAL, 'Cluj', 'Cluj-Napoca', 'Nicolae Titulescu', 14, '400420');

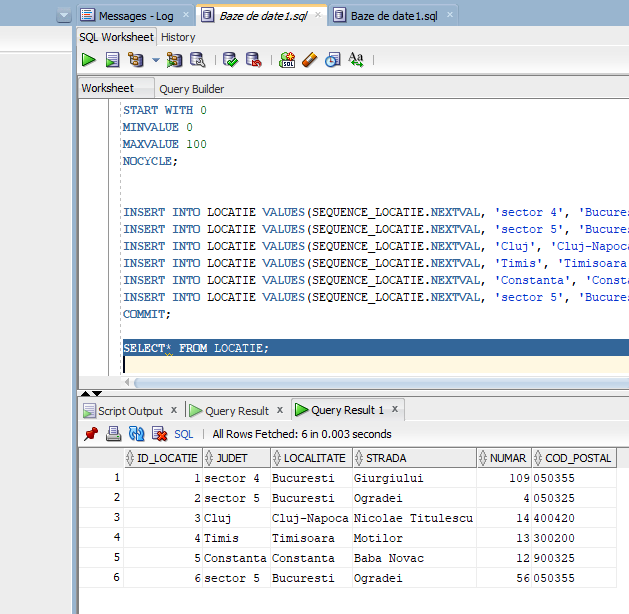
INSERT INTO LOCATIE VALUES(SEQUENCE\_LOCATIE.NEXTVAL, 'Timis', 'Timisoara', 'Motilor', 13, '300200');

INSERT INTO LOCATIE VALUES(SEQUENCE\_LOCATIE.NEXTVAL, 'Constanta', 'Constanta', 'Baba Novac', 12, '900325');

INSERT INTO LOCATIE VALUES(SEQUENCE\_LOCATIE.NEXTVAL, 'sector 5', 'Bucuresti', 'Ogradei', 56, '050355');

COMMIT;

SELECT\* FROM LOCATIE;



-------------------------

CREATE TABLE SERVICES

(id\_service number(6),

nume\_service varchar2(40) constraint nume\_service\_nn not null,

an\_infiintare number(4),

id\_locatie number(6),

constraint pk\_service primary key(id\_service),

constraint fk\_service\_locatie foreign key(id\_locatie) references LOCATIE(id\_locatie),

check (an\_infiintare > 1900)

);

CREATE SEQUENCE SEQUENCE\_SERVICE

START WITH 100

INCREMENT BY 1

MINVALUE 0

MAXVALUE 200

NOCYCLE;

INSERT INTO SERVICES VALUES(SEQUENCE\_SERVICE.NEXTVAL, 'Engine-fixers', 2005, 4);

INSERT INTO SERVICES VALUES(SEQUENCE\_SERVICE.NEXTVAL, 'Crazy-Wheel', 1998, 5);

INSERT INTO SERVICES VALUES(SEQUENCE\_SERVICE.NEXTVAL, 'ReadytoGo', 2015, 2);

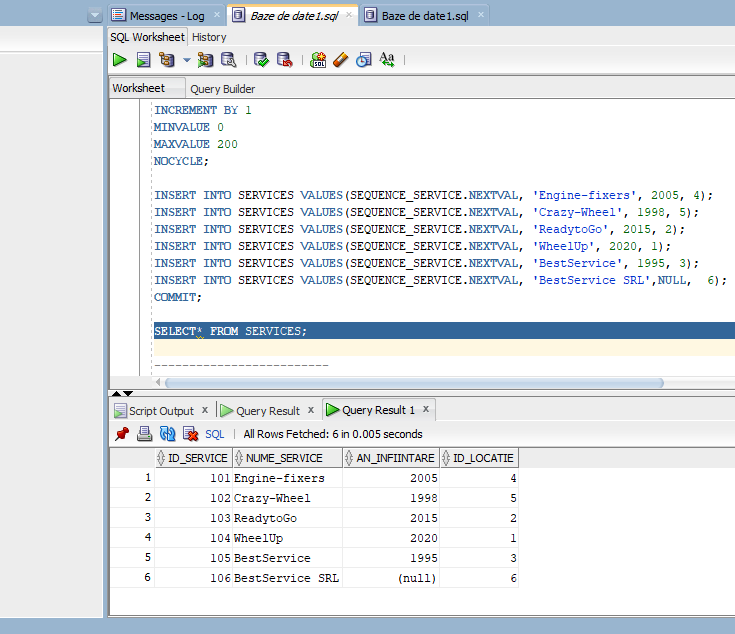
INSERT INTO SERVICES VALUES(SEQUENCE\_SERVICE.NEXTVAL, 'WheelUp', 2020, 1);

INSERT INTO SERVICES VALUES(SEQUENCE\_SERVICE.NEXTVAL, 'BestService', 1995, 3);

INSERT INTO SERVICES VALUES(SEQUENCE\_SERVICE.NEXTVAL, 'BestService SRL',NULL, 6);

COMMIT;

SELECT\* FROM SERVICES;



-------------------------

CREATE TABLE JOBS

(id\_job number(6),

nume\_job varchar2(30) constraint nume\_job\_nn not null,

salariu\_max number(6) constraint salariu\_max\_nn not null,

salariu\_min number(6) constraint salariu\_min\_nn not null,

constraint pk\_jobs primary key(id\_job)

);

INSERT INTO JOBS VALUES(500, 'Mecanic', 30000, 9000);

INSERT INTO JOBS VALUES(501, 'Tinichigiu', 20000, 5000);

INSERT INTO JOBS VALUES(502, 'Electrician auto', 25000, 6500);

INSERT INTO JOBS VALUES(503, 'Contabil', 22500, 9300);

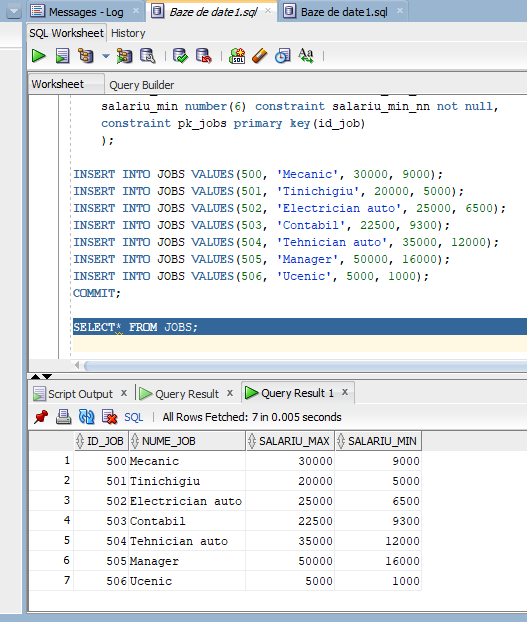
INSERT INTO JOBS VALUES(504, 'Tehnician auto', 35000, 12000);

INSERT INTO JOBS VALUES(505, 'Manager', 50000, 16000);

INSERT INTO JOBS VALUES(506, 'Ucenic', 5000, 1000);

COMMIT;

SELECT\* FROM JOBS;



-------------------------

CREATE TABLE ANGAJAT

(id\_angajat number(6),

nume\_angajat varchar2(30) constraint nume\_angajat\_nn not null,

prenume\_angajat varchar2(30) constraint prenume\_angajat\_nn not null,

salariu number(6) constraint salariu\_nn not null,

data\_angajarii date default sysdate,

id\_service number(6),

id\_job number(6),

constraint pk\_angajat primary key(id\_angajat),

constraint fk\_angajat\_service foreign key (id\_service) references SERVICES(id\_service),

constraint fk\_angajat\_job foreign key(id\_job) references JOBS(id\_job)

);

INSERT INTO ANGAJAT VALUES(1000, 'Vasile', 'Ionel', 12000, TO\_DATE('23/11/2009', 'DD/MM/YYYY'), 102, 500);

INSERT INTO ANGAJAT VALUES(1001, 'Ionescu', 'Gigel', 10000, TO\_DATE('28/10/2003', 'DD/MM/YYYY'), 102, 502);

INSERT INTO ANGAJAT VALUES(1002, 'Vasilescu', 'Ion', 11550, TO\_DATE('11/02/1999', 'DD/MM/YYYY'), 102, 503);

INSERT INTO ANGAJAT VALUES(1003, 'Popescu', 'Marin', 16000, TO\_DATE('19/07/2019','DD/MM/YYYY'), 101, 500);

INSERT INTO ANGAJAT VALUES(1004, 'Iordache', 'Adrian', 6000, TO\_DATE('23/05/2017', 'DD/MM/YYYY'), 101, 501);

INSERT INTO ANGAJAT VALUES(1005, 'Enescu', 'Alin', 42500, TO\_DATE('05/03/2007', 'DD/MM/YYYY'), 101, 505);

INSERT INTO ANGAJAT VALUES(1006, 'Deculescu', 'Emanuel', 10000, TO\_DATE('26/04/2016', 'DD/MM/YYYY'), 103, 503);

INSERT INTO ANGAJAT VALUES(1007, 'Mircescu', 'Ovidiu', 30000, TO\_DATE('28/08/2016', 'DD/MM/YYYY'), 103, 500);

INSERT INTO ANGAJAT VALUES(1008, 'Sandu', 'Mircea', 7000, TO\_DATE('21/12/2019', 'DD/MM/YYYY'), 103, 501);

INSERT INTO ANGAJAT VALUES(1009, 'Patrascu', 'Dorel', 18550, TO\_DATE('13/04/2021', 'DD/MM/YYYY'), 104, 504);

INSERT INTO ANGAJAT VALUES(1010, 'Davidescu', 'Filip', 10800, TO\_DATE('15/02/2021', 'DD/MM/YYYY'), 104, 503);

INSERT INTO ANGAJAT VALUES(1011, 'Isarescu', 'Iulian', 25000,TO\_DATE('29/11/2020', 'DD/MM/YYYY'), 104, 505);

INSERT INTO ANGAJAT VALUES(1012, 'Pascanu', 'Grigore', 1500,TO\_DATE('06/01/1996', 'DD/MM/YYYY'), 105, 506);

INSERT INTO ANGAJAT VALUES(1013, 'Niculai', 'Radu', 12000, TO\_DATE('20/05/2003', 'DD/MM/YYYY'), 105, 501);

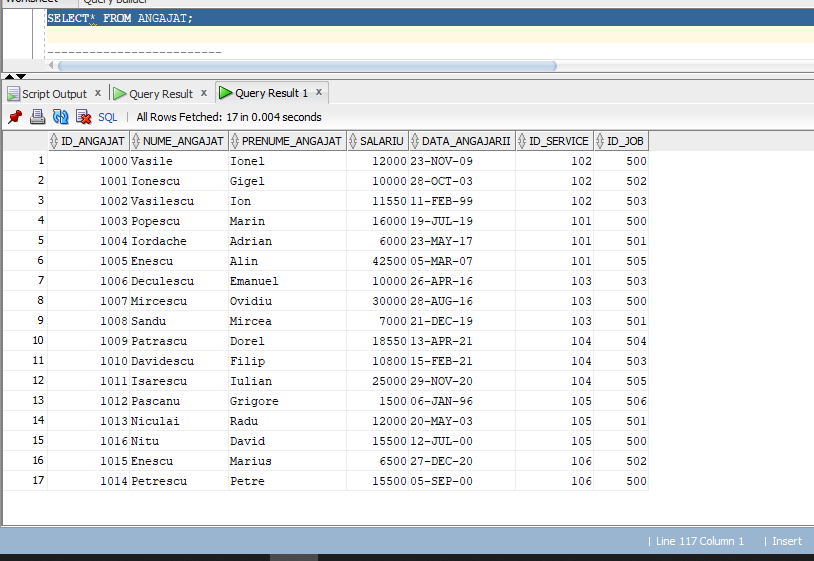
INSERT INTO ANGAJAT VALUES(1014, 'Petrescu', 'Petre', 15500, TO\_DATE('5/09/2000', 'DD/MM/YYYY'), 106, 500);

INSERT INTO ANGAJAT VALUES(1015, 'Enescu', 'Marius', 6500, TO\_DATE('27/12/2020', 'DD/MM/YYYY'), 106, 502);

INSERT INTO ANGAJAT VALUES(1016, 'Nitu', 'David', 15500, TO\_DATE('12/07/2000', 'DD/MM/YYYY'), 105, 500);

COMMIT;

SELECT \* FROM ANGAJAT;



-------------------------

CREATE TABLE CLIENTS

(id\_client number(6),

nume\_client varchar2(30) constraint nume\_client\_nn not null,

prenume\_client varchar2(30),

numar\_telefon varchar2(13),

constraint pk\_client primary key(id\_client)

);

INSERT INTO CLIENTS VALUES('10000', 'Raducanu', 'Andrei', '0712345678');

INSERT INTO CLIENTS VALUES('10001', 'Raducanu', 'Cosmin', '0712345679');

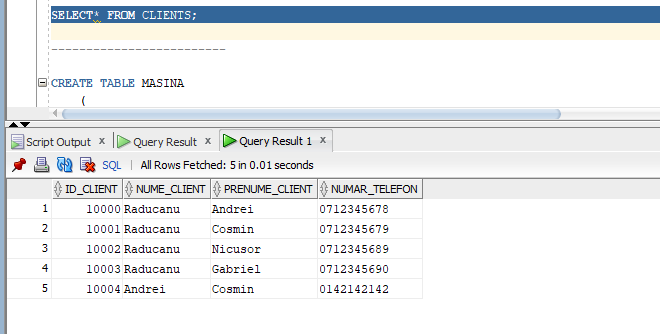
INSERT INTO CLIENTS VALUES('10002', 'Raducanu', 'Nicusor', '0712345689');

INSERT INTO CLIENTS VALUES('10003', 'Raducanu', 'Gabriel', '0712345690');

INSERT INTO CLIENTS VALUES('10004', 'Andrei', 'Cosmin', '0142142142');

COMMIT;

SELECT\* FROM CLIENTS;



-------------------------

CREATE TABLE MASINA

(

id\_masina number(6),

marca varchar(20) constraint marca\_nn not null,

an\_fabricatie number(4) constraint an\_fabricatie\_nn not null,

constraint pk\_masina primary key(id\_masina),

check(an\_fabricatie>1960)

);

INSERT INTO MASINA VALUES('10100', 'Audi', '2008');

INSERT INTO MASINA VALUES('10101', 'BMW', '2012');

INSERT INTO MASINA VALUES('10102', 'Mercedes-Benz', '2015');

INSERT INTO MASINA VALUES('10103', 'Skoda', '2006');

INSERT INTO MASINA VALUES('10104', 'Volkswagen', '1998');

INSERT INTO MASINA VALUES('10105', 'Afla Romeo', '2002');

INSERT INTO MASINA VALUES('10106', 'Ford', '2018');

INSERT INTO MASINA VALUES('10107', 'Fiat', '2017');

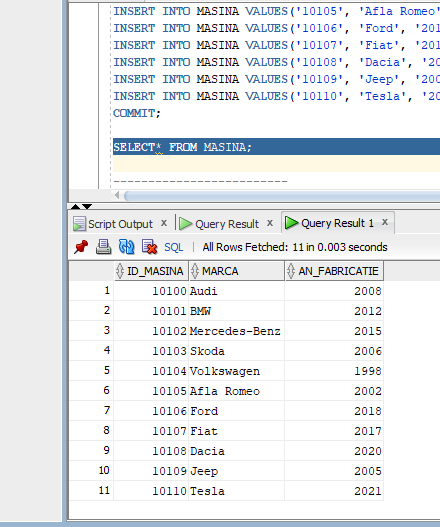
INSERT INTO MASINA VALUES('10108', 'Dacia', '2020');

INSERT INTO MASINA VALUES('10109', 'Jeep', '2005');

INSERT INTO MASINA VALUES('10110', 'Tesla', '2021');

COMMIT;

SELECT\* FROM MASINA;



-------------------------

CREATE TABLE PROGRAMARE

(id\_programare number(6),

data\_programare date default sysdate,

id\_client number(6),

constraint pk\_programare primary key(id\_programare),

constraint fk\_programare\_client foreign key(id\_client) references CLIENTS(id\_client)

);

INSERT INTO PROGRAMARE VALUES('100000', TO\_DATE('31/05/2021', 'DD/MM/YYYY'), '10000');

INSERT INTO PROGRAMARE VALUES('100001', TO\_DATE('21/06/2021', 'DD/MM/YYYY'), '10001');

INSERT INTO PROGRAMARE VALUES('100002', TO\_DATE('17/06/2021', 'DD/MM/YYYY'), '10000');

INSERT INTO PROGRAMARE VALUES('100003', TO\_DATE('15/07/2021', 'DD/MM/YYYY'), '10002');

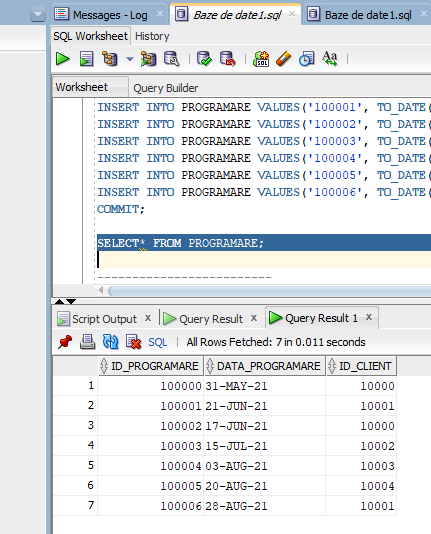
INSERT INTO PROGRAMARE VALUES('100004', TO\_DATE('03/08/2021', 'DD/MM/YYYY'), '10003');

INSERT INTO PROGRAMARE VALUES('100005', TO\_DATE('20/08/2021', 'DD/MM/YYYY'), '10004');

INSERT INTO PROGRAMARE VALUES('100006', TO\_DATE('28/08/2021', 'DD/MM/YYYY'), '10001');

COMMIT;

SELECT\* FROM PROGRAMARE;



-------------------------

CREATE TABLE CONTRACT

(id\_contract number(6),

data\_incepere date constraint data\_incepere\_nn not null,

durata number(6,2) constraint durata\_nn not null,

constraint pk\_contract primary key(id\_contract)

);

INSERT INTO CONTRACT VALUES('20000', TO\_DATE('22/07/2019', 'DD/MM/YYYY'), '3');

INSERT INTO CONTRACT VALUES('20001', TO\_DATE('18/09/2018', 'DD/MM/YYYY'), '3.5');

INSERT INTO CONTRACT VALUES('20002', TO\_DATE('12/11/2020', 'DD/MM/YYYY'), '1.25');

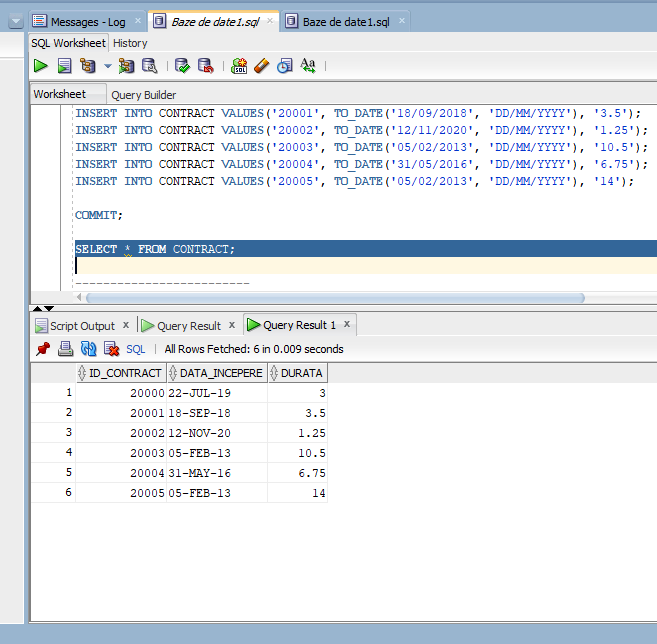
INSERT INTO CONTRACT VALUES('20003', TO\_DATE('05/02/2013', 'DD/MM/YYYY'), '10.5');

INSERT INTO CONTRACT VALUES('20004', TO\_DATE('31/05/2016', 'DD/MM/YYYY'), '6.75');

INSERT INTO CONTRACT VALUES('20005', TO\_DATE('05/02/2013', 'DD/MM/YYYY'), '14');

COMMIT;

SELECT \* FROM CONTRACT;



-------------------------

CREATE TABLE FURNIZOR

(id\_furnizor number(6),

nume\_furnizor varchar2(40) constraint nume\_furnizor\_nn not null,

valoare number(8) constraint valoare\_nn not null,

id\_contract number(6),

constraint pk\_furnizor primary key(id\_furnizor),

constraint fk\_furnizor\_contract foreign key(id\_contract) references CONTRACT(id\_contract)

);

INSERT INTO FURNIZOR VALUES('30000', 'Auto Rebel', '1000000', '20000');

INSERT INTO FURNIZOR VALUES('30001', 'Lipi Impex 2110', '1800000', '20004');

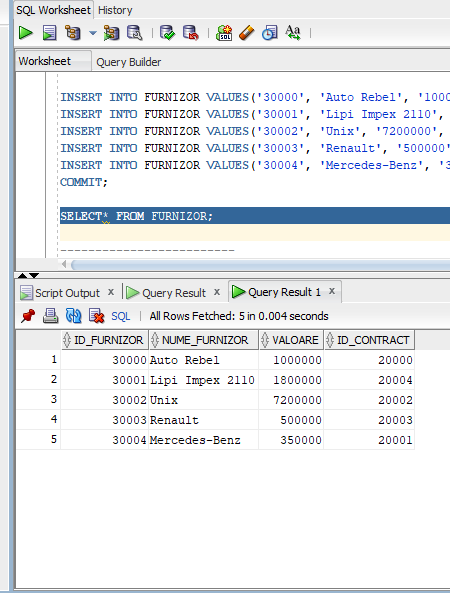
INSERT INTO FURNIZOR VALUES('30002', 'Unix', '7200000', '20002');

INSERT INTO FURNIZOR VALUES('30003', 'Renault', '500000', '20003');

INSERT INTO FURNIZOR VALUES('30004', 'Mercedes-Benz', '350000', '20001');

COMMIT;

SELECT\* FROM FURNIZOR;



-------------------------

CREATE TABLE PIESA

(id\_piesa number(6),

nume\_piesa varchar2(35)constraint nume\_piesa\_nn not null,

fabricata\_de varchar2(40) constraint fabricata\_de\_nn not null,

id\_furnizor number(6),

constraint pk\_piesa primary key(id\_piesa),

constraint fk\_piesa\_furnizor foreign key(id\_furnizor) references FURNIZOR(id\_furnizor)

);

INSERT INTO PIESA VALUES('40000', 'Bujie', 'Renault', '30004');

INSERT INTO PIESA VALUES('40001', 'Ambreiaj', 'Mercedes-Benz', '30002');

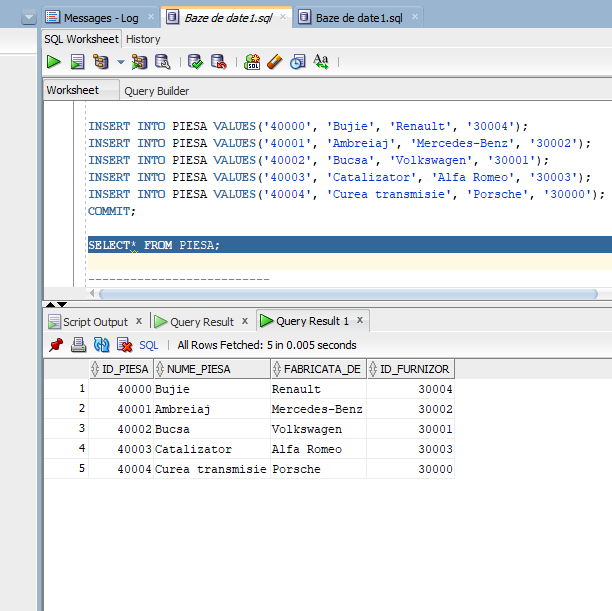
INSERT INTO PIESA VALUES('40002', 'Bucsa', 'Volkswagen', '30001');

INSERT INTO PIESA VALUES('40003', 'Catalizator', 'Alfa Romeo', '30003');

INSERT INTO PIESA VALUES('40004', 'Curea transmisie', 'Porsche', '30000');

COMMIT;

SELECT\* FROM PIESA;



--------------------------

CREATE TABLE MONTEAZA

(id\_angajat number(6),

id\_piesa number(6),

constraint pk\_monteaza primary key(id\_angajat, id\_piesa),

constraint fk\_monteaza\_angajat foreign key(id\_angajat) references ANGAJAT(id\_angajat),

constraint fk\_monteaza\_piesa foreign key(id\_piesa) references PIESA(id\_piesa)

);

INSERT INTO MONTEAZA VALUES('1000', '40000');

INSERT INTO MONTEAZA VALUES('1000', '40003');

INSERT INTO MONTEAZA VALUES('1001', '40002');

INSERT INTO MONTEAZA VALUES('1001', '40001');

INSERT INTO MONTEAZA VALUES('1002', '40002');

INSERT INTO MONTEAZA VALUES('1003', '40001');

INSERT INTO MONTEAZA VALUES('1005', '40003');

INSERT INTO MONTEAZA VALUES('1007', '40004');

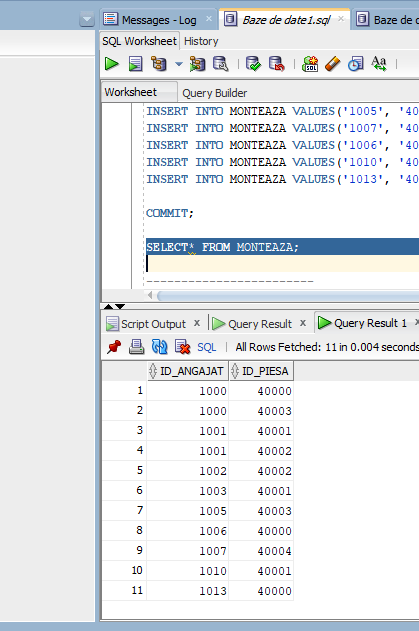
INSERT INTO MONTEAZA VALUES('1006', '40000');

INSERT INTO MONTEAZA VALUES('1010', '40001');

INSERT INTO MONTEAZA VALUES('1013', '40000');

COMMIT;

SELECT\* FROM MONTEAZA;



------------------------

CREATE TABLE REPARA

(id\_angajat number(6),

id\_masina number(6),

id\_client number(6),

constraint pk\_repara primary key(id\_angajat, id\_masina, id\_client),

constraint fk\_repara\_angajat foreign key(id\_angajat) references ANGAJAT(id\_angajat),

constraint fk\_repara\_client foreign key(id\_client) references CLIENTS(id\_client),

constraint fk\_repara\_masina foreign key(id\_masina) references MASINA(id\_masina));

INSERT INTO REPARA VALUES('1001', '10100', '10003');

INSERT INTO REPARA VALUES('1000', '10101', '10002');

INSERT INTO REPARA VALUES('1002', '10102', '10000');

INSERT INTO REPARA VALUES('1003', '10103', '10001');

INSERT INTO REPARA VALUES('1005', '10104', '10004');

INSERT INTO REPARA VALUES('1006', '10105', '10002');

INSERT INTO REPARA VALUES('1007', '10106', '10001');

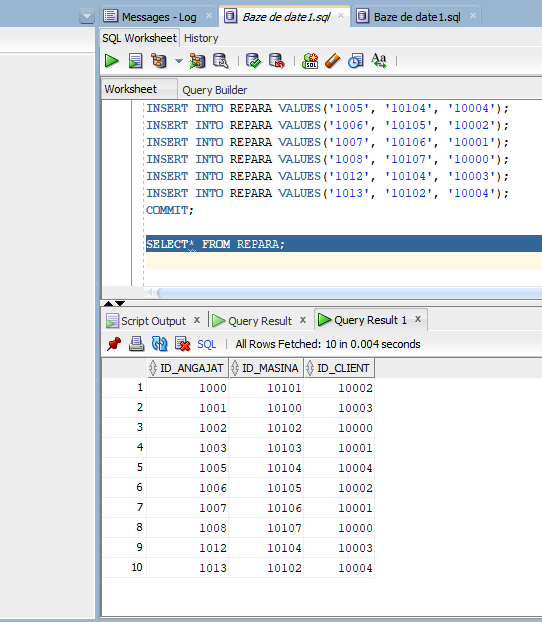
INSERT INTO REPARA VALUES('1008', '10107', '10000');

INSERT INTO REPARA VALUES('1012', '10104', '10003');

INSERT INTO REPARA VALUES('1013', '10102', '10004');

COMMIT;

SELECT\* FROM REPARA;



**Exercitiul 6.**

Afisati pe ecran codurile angajatilor unui service (ce are id-ul dat ca parametru) si care au montat cel putin o piesa care provine de la un furnizor ce a semnat un contract pe o durata strict mai mare de 3 ani.

Afisati intai codurile tuturor angajatilor si ale tuturor pieselor care respecta cerintele. Afisati apoi perechile id(angajat, piesa) care respecta cerinta.

CREATE OR REPLACE PROCEDURE angajat\_piesa (cod\_service services.id\_service%TYPE)

IS

TYPE tablou\_imbricat IS TABLE OF piesa.id\_piesa%TYPE;

tablou\_piese tablou\_imbricat := tablou\_imbricat();

TYPE vector\_angajati IS VARRAY(30) OF angajat.id\_angajat%TYPE;

vector\_ang vector\_angajati := vector\_angajati();

contor NUMBER(3) := 0;

ang monteaza.id\_angajat%TYPE;

pie monteaza.id\_piesa%TYPE;

v\_a NUMBER(1) := 0;

v\_p NUMBER(1) := 0;

BEGIN

SELECT id\_piesa

BULK COLLECT INTO tablou\_piese

FROM piesa JOIN furnizor USING(id\_furnizor)

JOIN contract USING(id\_contract)

WHERE durata > 3;

SELECT id\_angajat

BULK COLLECT INTO vector\_ang

FROM angajat

WHERE id\_service = cod\_service;

DBMS\_OUTPUT.PUT\_LINE('Codurile angajatilor care lucreaza la service-ul cu codul '|| cod\_service||' sunt: ');

FOR i IN vector\_ang.FIRST..vector\_ang.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE(vector\_ang(i));

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Codurile pieselor care provin de la furnizori ce au semnat un contract pe o durata mai mare de 3 ani sunt:');

FOR i IN tablou\_piese.FIRST..tablou\_piese.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE(tablou\_piese(i));

END LOOP;

SELECT COUNT(\*)

INTO contor

FROM monteaza;

FOR i IN 1..contor LOOP

v\_a := 0;

v\_p := 0;

SELECT id\_angajat, id\_piesa INTO ang, pie

FROM ( SELECT id\_angajat,id\_piesa, ROWNUM AS rn

FROM (SELECT id\_angajat, id\_piesa

FROM monteaza ORDER BY id\_angajat)) WHERE rn = i;

FOR j IN vector\_ang.FIRST..vector\_ang.LAST LOOP

IF vector\_ang(j) = ang THEN

v\_a := 1;

END IF;

END LOOP;

FOR k IN tablou\_piese.FIRST..tablou\_piese.LAST LOOP

IF tablou\_piese(k) = pie THEN

v\_p := 1;

END IF;

END LOOP;

IF v\_a = 1 AND v\_p = 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Angajatul cu codul ' || ang|| ' a montat piesa cu codul ' || pie);

END IF;

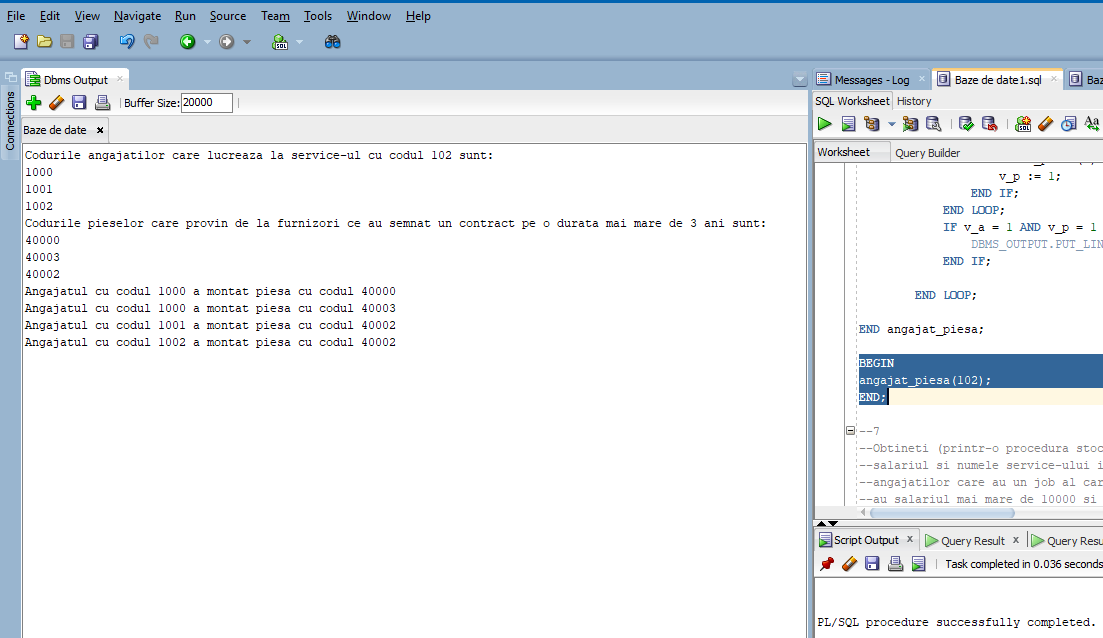
END LOOP;

END angajat\_piesa;

BEGIN

angajat\_piesa(102);

END;



**Exercitiul 7.**

Obtineti (printr-o procedura stocata) codul, numele, prenumele, salariul si numele service-ului in care lucreaza angajatii care au un job al carui nume este dat ca parametru, au salariul mai mare de 10000 si lucreaza intr-un service al carui nume incepe cu 'B'.

CREATE OR REPLACE PROCEDURE angajati\_service(nume jobs.nume\_job%TYPE)

IS

CURSOR c IS

(SELECT id\_angajat, nume\_angajat, prenume\_angajat, nume\_service, salariu

FROM angajat JOIN services USING(id\_service) JOIN jobs USING(id\_job)

WHERE UPPER(nume\_job) = UPPER(nume) AND salariu > 10000);

BEGIN

FOR i IN c LOOP

IF UPPER(i.nume\_service) LIKE 'B%' THEN

DBMS\_OUTPUT.PUT\_LINE('Angajatul cu codul ' ||i.id\_angajat|| ' si numele '||i.nume\_angajat||' '||i.prenume\_angajat|| ' are salariul ' || i.salariu ||' si lucreaza la service-ul '||i.nume\_service);

END IF;

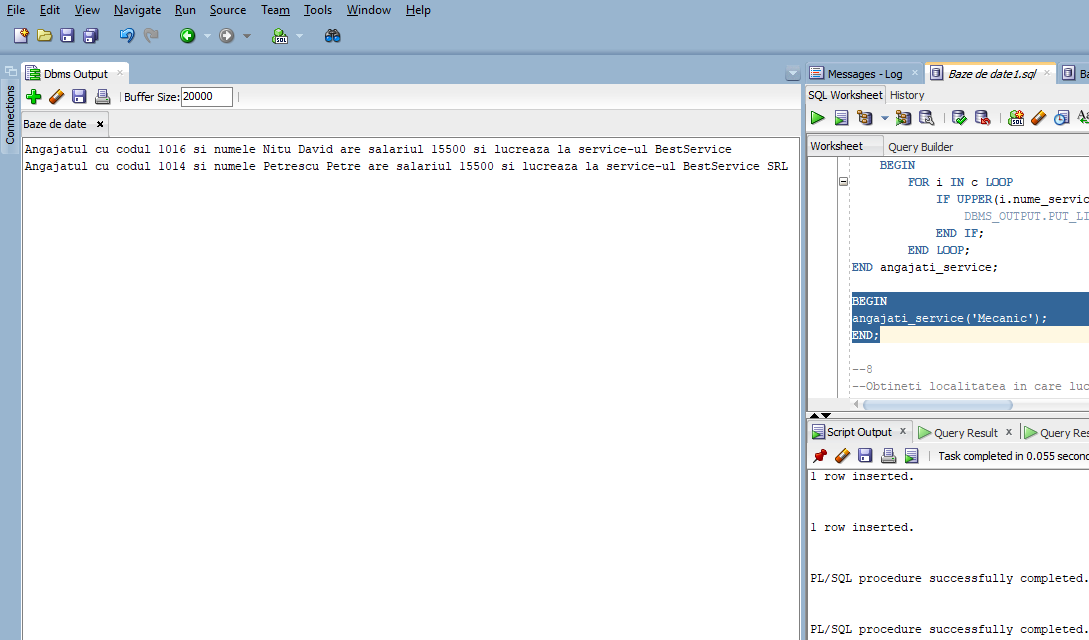
END LOOP;

END angajati\_service;

BEGIN

angajati\_service('Mecanic');

END;



**Exercitiul 8.**

Obtineti localitatea in care lucreaza un angajat al carui nume este dat ca parametru.

CREATE OR REPLACE FUNCTION oras\_angajat(nume angajat.nume\_angajat%TYPE)

RETURN locatie.localitate%TYPE IS

v\_localitate locatie.localitate%TYPE;

BEGIN

SELECT localitate INTO v\_localitate

FROM angajat JOIN services USING(id\_service) JOIN locatie USING(id\_locatie)

WHERE UPPER(nume\_angajat) = UPPER(nume);

RETURN v\_localitate;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista niciun angajat cu acest nume!');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi angajati cu acest nume!');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'A aparut alta eroare!');

END oras\_angajat;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(oras\_angajat('Mircescu')); -- nicio exceptie

END;

BEGIN

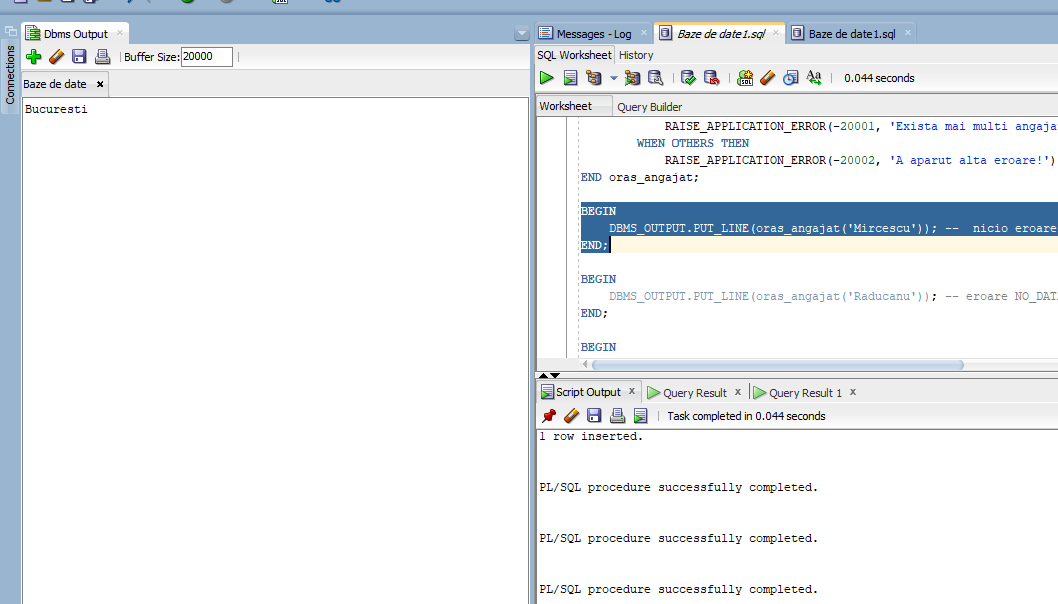
DBMS\_OUTPUT.PUT\_LINE(oras\_angajat('Raducanu')); -- NO\_DATA\_FOUND

END;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(oras\_angajat('Enescu')); -- TOO\_MANY\_ROWS

END;



**Exercitiul 9.**

Obtineti data inceperii contractului prin care se furnizeaza piesa al carui nume este dat ca parametru si care este montata de cel mai bine platit angajat dintre cei care au montat acel tip de piesa.

CREATE OR REPLACE PROCEDURE piesa\_angajat (v\_nume\_piesa piesa.nume\_piesa%TYPE)

IS

v\_data\_incepere contract.data\_incepere%TYPE;

v\_max\_salariu angajat.salariu%TYPE;

BEGIN

SELECT MAX(salariu) INTO v\_max\_salariu

FROM angajat JOIN monteaza USING(id\_angajat)

JOIN piesa USING(id\_piesa)

WHERE UPPER(nume\_piesa) = UPPER(v\_nume\_piesa);

SELECT data\_incepere INTO v\_data\_incepere

FROM contract JOIN furnizor USING(id\_contract)

JOIN piesa USING(id\_furnizor)

JOIN monteaza USING(id\_piesa)

JOIN angajat USING(id\_angajat)

WHERE UPPER(nume\_piesa) = UPPER(v\_nume\_piesa) AND salariu = v\_max\_salariu;

DBMS\_OUTPUT.PUT\_LINE('Data cautata este: '||v\_data\_incepere);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista angajat care sa fi montat piesa cu acest nume!');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista prea multi angajati care au salariul maxim si au montat aceasta piesa!');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'A aparut alta eroare!');

END piesa\_angajat;

BEGIN

piesa\_angajat('acceleratie'); -- NO\_DATA\_FOUND

END;

BEGIN

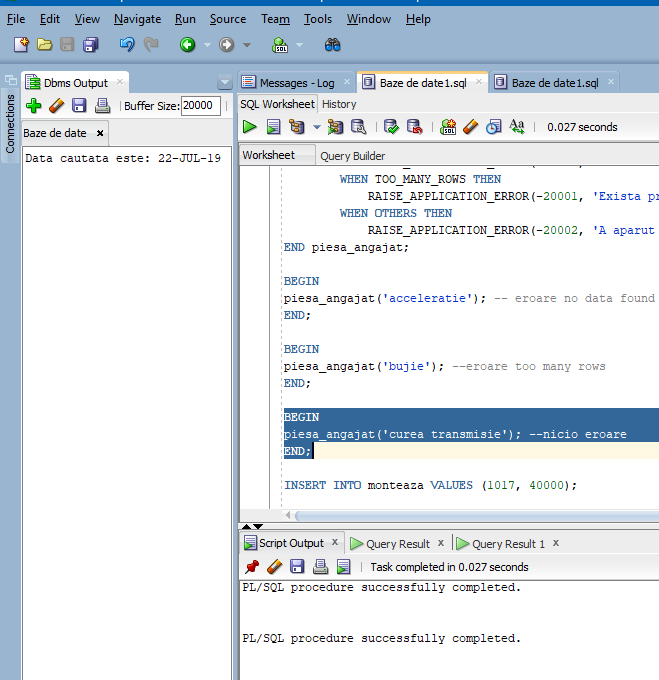
piesa\_angajat('bujie'); -- TOO\_MANY\_ROWS

END;

BEGIN

piesa\_angajat('curea transmisie'); --nicio exceptie

END;



**Exercitiul 10.**

Creati un trigger prin care sa nu se poata modifica tabelul 'angajat' decat intre orele 8-16 de luni pana vineri.

CREATE OR REPLACE TRIGGER actualizare\_ang

BEFORE INSERT OR UPDATE OR DELETE ON angajat

BEGIN

IF (TO\_CHAR(SYSDATE,'D') = 7) OR (TO\_CHAR(SYSDATE,'D') = 6) OR (TO\_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 16)THEN

IF INSERTING THEN

RAISE\_APPLICATION\_ERROR(-20001,'Nu se poate insera in tabel in afara programului');

ELSIF UPDATING THEN

RAISE\_APPLICATION\_ERROR(-20002,'Nu se poate actualiza tabelul in afara programului');

ELSE

RAISE\_APPLICATION\_ERROR(-20003,'Nu se poate sterge din tabel in afara programului');

END IF;

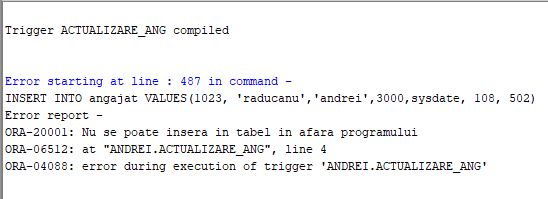
END IF;

END;

--INSERT INTO angajat VALUES(1023, 'raducanu','andrei',3000,sysdate, 108, 502);

--UPDATE angajat SET salariu = salariu/2;

--DELETE FROM angajat WHERE id\_angajat = 1005;



**Exercitiul 11.**

Creati un trigger prin care sa nu se poata adauga un anagajat care are salariul necorespunzator grilei de salarizare specifica job-ului avut.

CREATE OR REPLACE TRIGGER salariu\_angajat

BEFORE INSERT ON angajat

FOR EACH ROW

DECLARE

v\_sal\_min jobs.salariu\_min%TYPE;

v\_sal\_max jobs.salariu\_max%TYPE;

BEGIN

v\_sal\_min := -1;

v\_sal\_max := -1;

SELECT salariu\_min, salariu\_max

INTO v\_sal\_min, v\_sal\_max

FROM JOBS

WHERE id\_job = :NEW.id\_job;

IF :NEW.salariu < v\_sal\_min OR :NEW.salariu > v\_sal\_max THEN

IF :NEW.salariu < v\_sal\_min THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Salariul dat este mai mic decat limita inferioara pentru acest job!');

ELSE

RAISE\_APPLICATION\_ERROR(-20002, 'Salariul dat este mai mare decat limita superioara pentru acest job!');

END IF;

END IF;

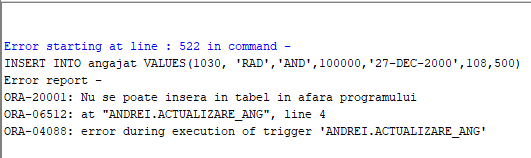
EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Nu exista acest cod pentru tabela job, dati un id valid!');

END;

--INSERT INTO angajat VALUES(1030, 'RAD','AND',100000,'27-DEC-2000',108,500);



**Exercitiul 12.**

Creati un trigger LDD ce retine detalii in tabelul audit\_ara de fiecare data cand este facuta o comanda CREATE, ALTER, DROP.

CREATE TABLE audit\_ara

(utilizator VARCHAR2(30),

baza\_date VARCHAR2(50),

eveniment VARCHAR2(20),

nume\_obiect VARCHAR2(30),

data DATE);

CREATE OR REPLACE TRIGGER ldd\_trigger

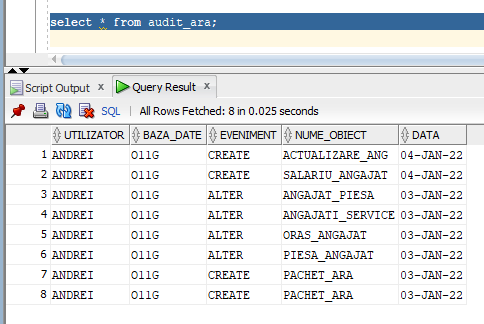
AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO audit\_ara VALUES

(SYS.LOGIN\_USER, SYS.DATABASE\_NAME, SYS.SYSEVENT, SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);

END;



**Exercitiul 13**

Creati un pachet ce contine toate procedurile si functiile definite in proiect.

CREATE OR REPLACE PACKAGE pachet\_ara AS

PROCEDURE angajat\_piesa(cod\_service services.id\_service%TYPE);

PROCEDURE angajati\_service(nume jobs.nume\_job%TYPE);

FUNCTION oras\_angajat(nume angajat.nume\_angajat%TYPE)

RETURN locatie.localitate%TYPE;

PROCEDURE piesa\_angajat(v\_nume\_piesa piesa.nume\_piesa%TYPE);

END;

CREATE OR REPLACE PACKAGE BODY pachet\_ara AS

--6

PROCEDURE angajat\_piesa(cod\_service services.id\_service%TYPE)

IS

TYPE tablou\_imbricat IS TABLE OF piesa.id\_piesa%TYPE;

tablou\_piese tablou\_imbricat := tablou\_imbricat();

TYPE vector\_angajati IS VARRAY(30) OF angajat.id\_angajat%TYPE;

vector\_ang vector\_angajati := vector\_angajati();

contor NUMBER(3) := 0;

ang monteaza.id\_angajat%TYPE;

pie monteaza.id\_piesa%TYPE;

v\_a NUMBER(1) := 0;

v\_p NUMBER(1) := 0;

BEGIN

SELECT id\_piesa

BULK COLLECT INTO tablou\_piese

FROM piesa JOIN furnizor USING(id\_furnizor)

JOIN contract USING(id\_contract)

WHERE durata > 3;

SELECT id\_angajat

BULK COLLECT INTO vector\_ang

FROM angajat

WHERE id\_service = cod\_service;

DBMS\_OUTPUT.PUT\_LINE('Codurile angajatilor care lucreaza la service-ul cu codul '|| cod\_service||' sunt: ');

FOR i IN vector\_ang.FIRST..vector\_ang.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE(vector\_ang(i));

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Codurile pieselor care provin de la furnizori ce au semnat un contract pe o durata mai mare de 3 ani sunt:');

FOR i IN tablou\_piese.FIRST..tablou\_piese.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE(tablou\_piese(i));

END LOOP;

SELECT COUNT(\*)

INTO contor

FROM monteaza;

FOR i IN 1..contor LOOP

v\_a := 0;

v\_p := 0;

SELECT id\_angajat, id\_piesa INTO ang, pie

FROM ( SELECT id\_angajat,id\_piesa, ROWNUM AS rn

FROM (SELECT id\_angajat, id\_piesa

FROM monteaza ORDER BY id\_angajat)) WHERE rn = i;

FOR j IN vector\_ang.FIRST..vector\_ang.LAST LOOP

IF vector\_ang(j) = ang THEN

v\_a := 1;

END IF;

END LOOP;

FOR k IN tablou\_piese.FIRST..tablou\_piese.LAST LOOP

IF tablou\_piese(k) = pie THEN

v\_p := 1;

END IF;

END LOOP;

IF v\_a = 1 AND v\_p = 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Angajatul cu codul ' || ang|| ' a montat piesa cu codul ' || pie);

END IF;

END LOOP;

END angajat\_piesa;

--7

PROCEDURE angajati\_service(nume jobs.nume\_job%TYPE)

IS

CURSOR c IS

(SELECT id\_angajat, nume\_angajat, prenume\_angajat, nume\_service, salariu

FROM angajat JOIN services USING(id\_service) JOIN jobs USING(id\_job)

WHERE UPPER(nume\_job) = UPPER(nume) AND salariu > 10000);

BEGIN

FOR i IN c LOOP

IF UPPER(i.nume\_service) LIKE 'B%' THEN

DBMS\_OUTPUT.PUT\_LINE('Angajatul cu codul ' ||i.id\_angajat|| ' si numele '||i.nume\_angajat||' '||i.prenume\_angajat|| ' are salariul ' || i.salariu ||' si lucreaza la service-ul '||i.nume\_service);

END IF;

END LOOP;

END angajati\_service;

--8

FUNCTION oras\_angajat(nume angajat.nume\_angajat%TYPE)

RETURN locatie.localitate%TYPE IS

v\_localitate locatie.localitate%TYPE;

BEGIN

SELECT localitate INTO v\_localitate

FROM angajat JOIN services USING(id\_service) JOIN locatie USING(id\_locatie)

WHERE UPPER(nume\_angajat) = UPPER(nume);

RETURN v\_localitate;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista niciun angajat cu acest nume!');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi angajati cu acest nume!');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'A aparut alta eroare!');

END oras\_angajat;

--9

PROCEDURE piesa\_angajat(v\_nume\_piesa piesa.nume\_piesa%TYPE)

IS

v\_data\_incepere contract.data\_incepere%TYPE;

v\_max\_salariu angajat.salariu%TYPE;

BEGIN

SELECT MAX(salariu) INTO v\_max\_salariu

FROM angajat JOIN monteaza USING(id\_angajat)

JOIN piesa USING(id\_piesa)

WHERE UPPER(nume\_piesa) = UPPER(v\_nume\_piesa);

SELECT data\_incepere INTO v\_data\_incepere

FROM contract JOIN furnizor USING(id\_contract)

JOIN piesa USING(id\_furnizor)

JOIN monteaza USING(id\_piesa)

JOIN angajat USING(id\_angajat)

WHERE UPPER(nume\_piesa) = UPPER(v\_nume\_piesa) AND salariu = v\_max\_salariu;

DBMS\_OUTPUT.PUT\_LINE('Data cautata este: '||v\_data\_incepere);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista angajat care sa fi montat piesa cu acest nume!');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista prea multi angajati care au salariul maxim si au montat aceasta piesa!');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'A aparut alta eroare!');

END piesa\_angajat;

END;

BEGIN

pachet\_ara.angajat\_piesa(102);

END;

BEGIN

pachet\_ara.angajati\_service('Mecanic');

END;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(oras\_angajat('Nitu'));

END;

BEGIN

piesa\_angajat('ambreiaj');

END;

