

Gestiunea unei societăți de transport

Sisteme de Gestiune a Bazelor de Date

Anul II, Sem. 1

Bercaru Alexandru-Mihai

Seria 23, Grupa 231

Anul Universitar 2024-2025

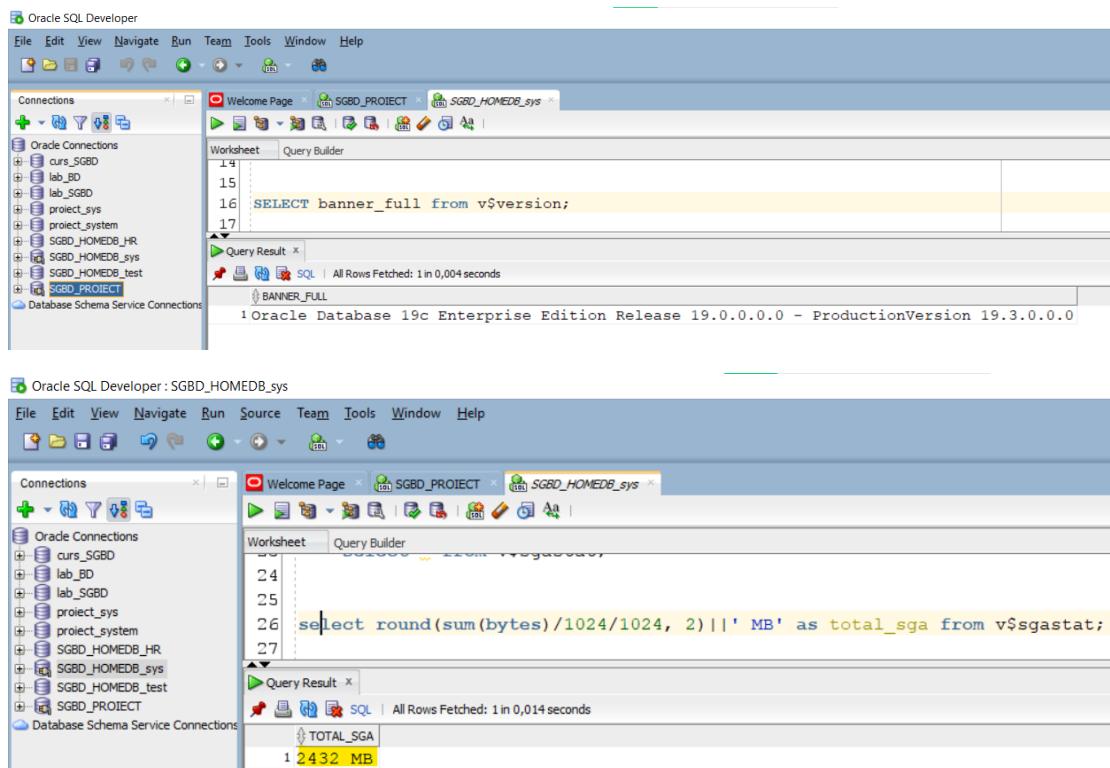
Cuprins

Introducere	3
Cerința 1: Prezentarea bazei de date	5
Cerința 2: Diagrama Entitate-Relație	6
Cerința 3: Diagrama Conceptuală.....	7
Cerința 4: Definirea tabelelor și a constrângerilor.....	8
Cerința 5: Inserarea de informații în tabele	12
Cerința 6: Colecții.....	35
Cerința 7: Cursoare	40
Cerința 8: Subprogram stocat de tip funcție, tratarea excepțiilor	44
Cerința 9: Subprogram stocat de tip procedură, excepții proprii.....	50
Cerința 10: Trigger LMD la nivel de comandă	58
Cerința 11: Trigger LMD la nivel de linie.....	60
Cerința 12: Trigger LDD	68
Cerința 13: Pachet.....	70

Introducere

Am ales să modelez o bază de date ce se ocupă cu administrarea unei societăți de transport în comun, informațiile de interes fiind legate de funcționarea internă a societății, în special de gestiunea vehiculelor, a traseelor pe care circulă acestea, a reparațiilor prin care trec sau a incidentelor în care sunt implicate.

Baza de date a fost implementată folosind SGBD-ul **Oracle Database 19c Enterprise Edition**, instalat pe sistemul de operare Microsoft Windows 10. La instalare, au fost alocate 2 GB RAM pentru funcționare.



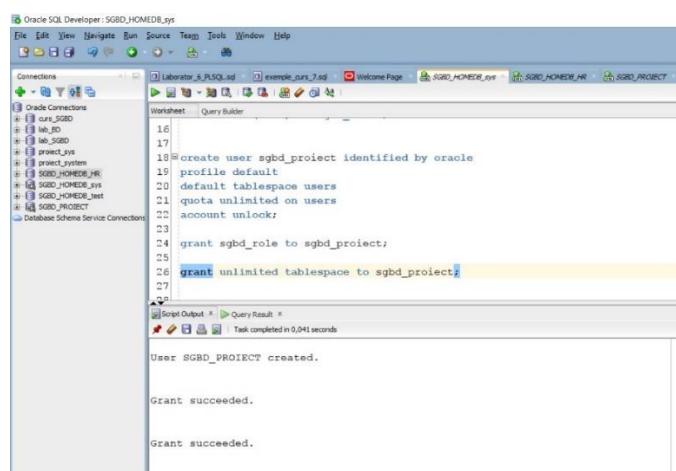
```

14
15
16 SELECT banner_full from v$version;
17

24
25
26 select round(sum(bytes)/1024/1024, 2) || ' MB' as total_sga from v$sgastat;
27

```

Pentru a implementa tabelele bazei de date, am creat un nou utilizator, fără drepturi de administrare, denumit **sgbd_project** (implicit, se va crea și schema aferentă acestui utilizator):



```

16 @create user sgbd_project identified by oracle
17 profile default
18 default tablespace users
19 quota unlimited on users
20 account unlock;
21
22 grant sgbd_role to sgbd_project;
23
24 grant unlimited tablespace to sgbd_project;
25
26

```

Script Output:

```

User SGBD_PROJECT created.

Grant succeeded.

Grant succeeded.

```

Rolul **sgbd_role** avea următoarele privilegii:

```

12
13 select *
14 from role_sys_privs
15 where lower(role) = 'sgbd_role';

```

ROLE	PRIVILEGE	ADMIN_OPTION	COMMON	INHERITED
1 SGBD_ROLE	CREATE SYNONYM	NO	YES	NO
2 SGBD_ROLE	COPY Rewrite	NO	YES	NO
3 SGBD_ROLE	CREATE PROCEDURE	NO	YES	NO
4 SGBD_ROLE	CREATE PUBLIC DATABASE LINK	NO	YES	NO
5 SGBD_ROLE	CREATE TABLE	NO	YES	NO
6 SGBD_ROLE	CREATE VIEW	NO	YES	NO
7 SGBD_ROLE	CREATE MATERIALIZED VIEW	NO	YES	NO
8 SGBD_ROLE	SELECT ANY DICTIONARY	NO	YES	NO
9 SGBD_ROLE	CREATE TRIGGER	NO	YES	NO
10 SGBD_ROLE	CREATE PUBLIC SYNONYM	NO	YES	NO
11 SGBD_ROLE	ALTER SESSION	NO	YES	NO
12 SGBD_ROLE	CREATE TYPE	NO	YES	NO
13 SGBD_ROLE	CREATE SEQUENCE	NO	YES	NO

Putem verifica faptul că, într-adevăr, rolul **sgbd_role** a fost acordat userului **sgbd_project**:

```

1 select * from user_role_privs;

```

USERNAME	GRANTED_ROLE	ADMIN_OPTION	DELEGATE_OPTION	DEFAULT_ROLE	OS_GRANTED	COMMON	INHERITED
1 SGBD_PROJECT	SGBD_ROLE	NO	NO	YES	NO	YES	NO

Cerință 1: Prezentarea bazei de date

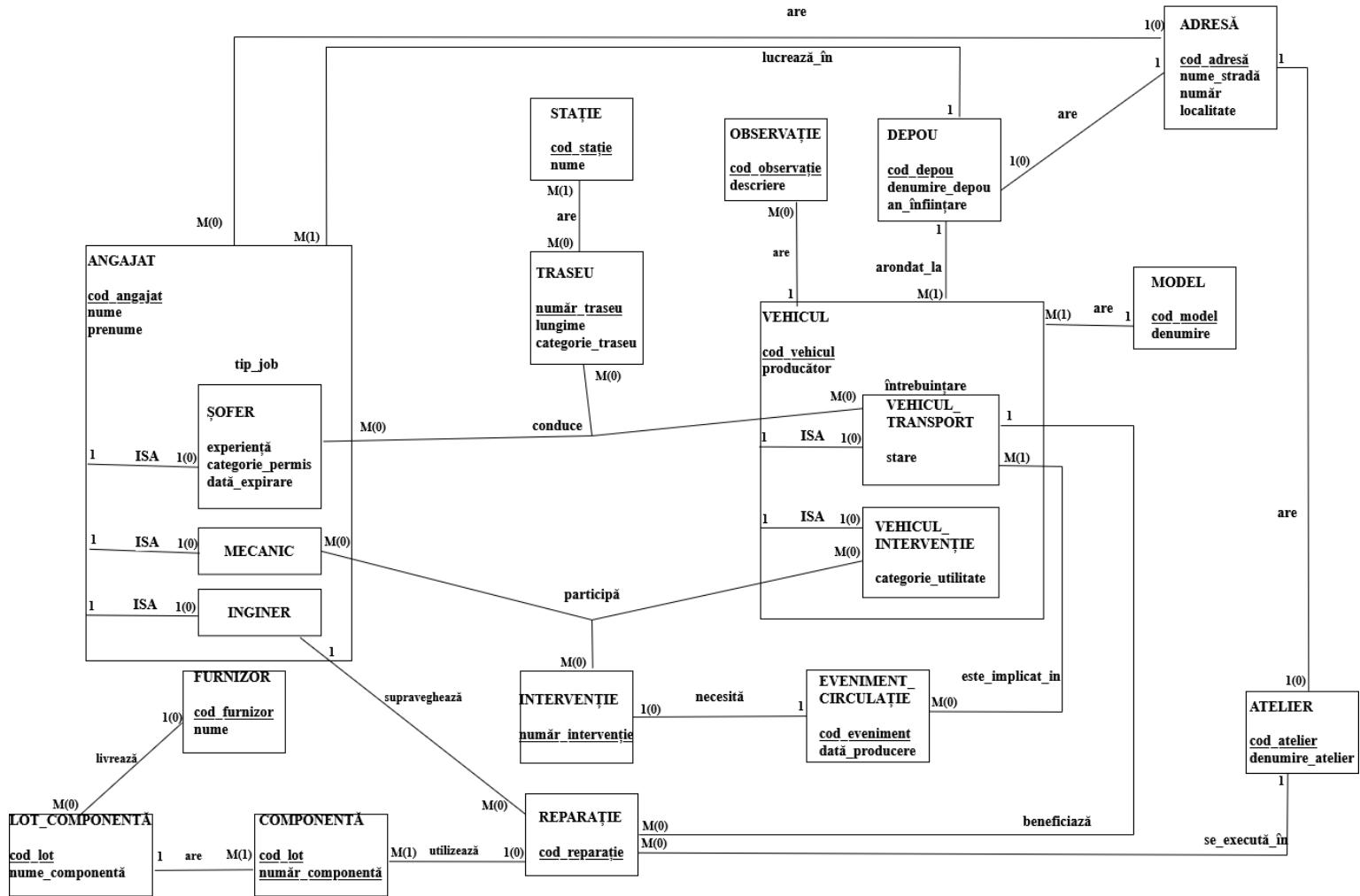
O societate de transport în comun, ce deservește un oraș și împrejurimile sale, operează mai multe trasee. Societatea deține un parc de vehicule, alcătuit din mijloace de transport (autobuze, tramvaie, troleibuze) și din vehicule de intervenție. Fiecare vehicul este arondat unui depou, în cadrul căruia lucrează angajați ai companiei. Pentru a ieși pe un traseu, șoferilor le sunt repartizate vehicule arondate depoului de care aparțin, doar din categoria de mijloc de transport ce corespunde cu categoria de permis de conducere (exemplu: vatmanilor le pot fi repartizate doar tramvaie, etc.). În decurs de 24 de ore, un șofer poate efectua maxim o singură cursă (pe timp de zi sau pe timp de noapte). Traseele sunt alcătuite din stații, iar acestea pot apărea pe mai multe linii.

Periodic, au loc evenimente de circulație, în urma căroror vehiculele pot suferi stricăciuni. Evenimentele de circulație pot fi atât accidente (coliziuni cu alte autovehicule, deraieri, etc.), cât și defecțiuni ale motoarelor sau altor componente. În cazul producerii acestor evenimente, o echipă de mecanici este trimisă în intervenție pentru a soluționa problema, folosind vehiculele destinate acestui scop. Uneori, sunt necesare și intervenții la infrastructură (șinele de tramvai, linii de contact pentru furnizarea curentului etc.), caz în care se procedează la fel. Vehiculele defecte sau învechite sunt trimise pentru reparații la un atelier, sub coordonarea unui inginer. În cadrul unei reparații, se pot înlocui mai multe componente. Un anumit model de componentă este cumpărat în loturi de la un furnizor.

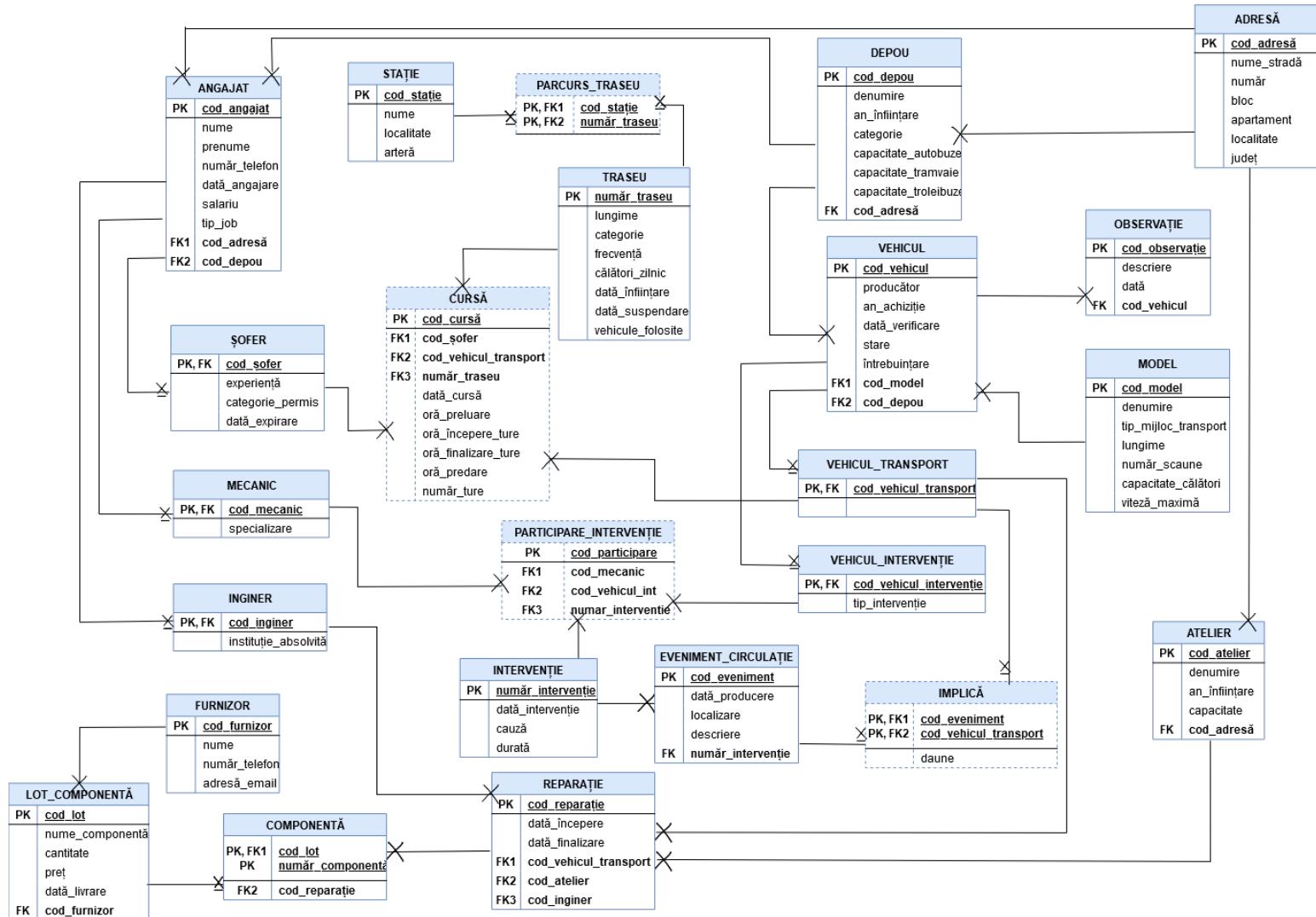
Printre constrângerile impuse asupra modelului se regăsesc:

- stațiile unui traseu sunt aceleași în ambele sensuri de circulație (tur și retur);
- fiecare vehicul trebuie să fie arondat la un depou și numai unul;
- un mijloc de transport poate trece printr-o singură reparație la un moment dat;
- un șofer poate avea un singur tip de permis de conducere (permis pentru autobuz, troleibuz sau tramvai) și poate fi repartizat doar pe un vehicul din categoria specificată în permis;
- pentru contactarea salariaților, s-a luat în calcul un singur număr de telefon;
- un vehicul de transport poate fi implicat în mai multe evenimente de circulație, care sunt produse în zile diferite;
- nu pot exista două stații cu același nume;
- se presupune că șoferilor le pot fi repartizate doar vehicule garate în depoul la care sunt arondați și aceștia;
- fiecare eveniment de circulație în care este implicat un vehicul al companiei necesită o intervenție, însă nu fiecare intervenție apare ca urmare a unui eveniment de circulație;
- fiecare lot livrat de un furnizor conține un singur tip de componentă;
- vehiculele de intervenție pot fi utilizate doar de către mecanici, în contextul necesității unei intervenții.
- același tip de componentă poate fi livrată de mai mulți furnizori și se presupune că piesa respectivă este compatibilă cu fiecare vehicul de o anumită categorie, indiferent de modelul acestuia.

Cerință 2: Diagrama Entitate-Relație



Cerință 3: Diagrama Conceptuală



Cerință 4: Definirea tabelelor și a constrângerilor

```

CREATE TABLE ADRESA(
cod_adresa      NUMBER(5)      CONSTRAINT pk_adresa PRIMARY KEY,
nume_strada     VARCHAR2(50)   NOT NULL,
numar           VARCHAR2(10)   NOT NULL,
bloc            VARCHAR2(10),
apartament      NUMBER(5),
localitate       VARCHAR2(20)   NOT NULL,
judet           VARCHAR2(20));

```

```

CREATE TABLE FURNIZOR(
cod_furnizor    NUMBER(3)      CONSTRAINT pk_furnizor PRIMARY KEY,
nume             VARCHAR2(50)   NOT NULL,
numar_telefon   CHAR(12)      NOT NULL,
adresa_email    VARCHAR2(50));

```

```

CREATE TABLE MODEL(
cod_model        NUMBER(5)      CONSTRAINT pk_model PRIMARY KEY,
denumire         VARCHAR2(20)   UNIQUE NOT NULL,
tip_mijloc_transport VARCHAR2(20) CHECK (tip_mijloc_transport IN
('tramvai', 'autobuz', 'troleibuz')),
lungime          NUMBER(2),
numar_scaune    NUMBER(3),
capacitate_calatori NUMBER(3),
viteza_maxima   NUMBER(5,2));

```

```

CREATE TABLE TRASEU(
numar_traseu    NUMBER(3)      CONSTRAINT pk_traseu PRIMARY
KEY,
lungime          NUMBER(5,2)    NOT NULL,
categorie        VARCHAR2(15)   NOT NULL CHECK(categorie IN
('urban', 'regional', 'de noapte')),
frecventa       INTERVAL DAY TO SECOND NOT NULL,
medie_calatori  NUMBER(6)      NOT NULL,
data_infiintare DATE          NOT NULL,
data_suspendare DATE);

```

```

CREATE TABLE STATIE(
cod_statie       NUMBER(5)      CONSTRAINT pk_statie PRIMARY KEY,
nume             VARCHAR2(25)   NOT NULL UNIQUE,
localitate       VARCHAR2(25)   NOT NULL,
artera           VARCHAR2(100)  NOT NULL);

```

```

CREATE TABLE INTERVENTIE(
numar_interventie NUMBER(5)      CONSTRAINT pk_interventie
PRIMARY KEY,
data_interventie  TIMESTAMP WITH TIME ZONE NOT NULL,
cauza            VARCHAR2(100)  NOT NULL,
durata           INTERVAL DAY TO SECOND NOT NULL);

```

```

CREATE TABLE DEPOU(
cod_depou        NUMBER(5)      CONSTRAINT pk_depou PRIMARY KEY,
denumire         VARCHAR2(50)   UNIQUE NOT NULL,
an_infiintare   DATE          NOT NULL,
categorie        VARCHAR2(50)   NOT NULL CHECK(categorie IN('autobaza',
'depou tramvaie', 'depou troleibuze', 'depou mixt')),
capacitate_autobuze NUMBER(3),
capacitate_troleibuze NUMBER(3),
capacitate_tramvaie NUMBER(3),
cod_adresa       NUMBER(5)
CONSTRAINT fk_depou_adr REFERENCES ADRESA(cod_adresa) ON DELETE SET NULL,
CONSTRAINT adr_depou_unic UNIQUE(cod_adresa));

```

```

CREATE TABLE VEHICUL(
cod_vehicul      NUMBER(5)      CONSTRAINT pk_vehicul PRIMARY KEY,

```

```

producator      VARCHAR2(25)      NOT NULL,
an_achizitie    DATE            NOT NULL,
data_verificare DATE            DEFAULT TO_DATE('06/01/2025', 'dd/mm/yyyy'),
stare          VARCHAR2(20)     DEFAULT 'functional'
                                         CONSTRAINT verif_stare CHECK(stare
IN('functional', 'defect', 'in reparatie')),
kilometraj     NUMBER(10)       DEFAULT 0,
intrebuintare  VARCHAR2(20)     CHECK(intrebuintare IN('transport',
'interventie')),
cod_depou      NUMBER(5)        NOT NULL,
cod_model      NUMBER(5),
CONSTRAINT fk_veh_dep FOREIGN KEY (cod_depou) REFERENCES DEPOU(cod_depou) ON
DELETE SET NULL,
CONSTRAINT fk_veh_mod FOREIGN KEY (cod_model) REFERENCES MODEL(cod_model) ON
DELETE SET NULL);

CREATE TABLE VEHICUL_TRANSPORT(
cod_vehicul_tr  NUMBER(5)      CONSTRAINT pk_veh_tr PRIMARY KEY,
CONSTRAINT fk_veh_tr FOREIGN KEY (cod_vehicul_tr) REFERENCES VEHICUL(cod_vehicul)
ON DELETE CASCADE);

CREATE TABLE VEHICUL_INTERVENTIE(
cod_vehicul_int NUMBER(5)      CONSTRAINT pk_veh_int PRIMARY KEY,
categorie_utilitate VARCHAR2(50) NOT NULL,
CONSTRAINT fk_veh_int FOREIGN KEY(cod_vehicul_int) REFERENCES
VEHICUL(cod_vehicul) ON DELETE CASCADE);

CREATE TABLE PARCURS_TRASEU(
numar_traseu    NUMBER(3)      CONSTRAINT fk_parcurs_trs REFERENCES
TRASEU(numar_traseu) ON DELETE CASCADE,
cod_statie      NUMBER(5)      CONSTRAINT fk_parcurs_stat REFERENCES
STATIE(cod_statie) ON DELETE CASCADE,
numar_ordine    NUMBER(3),
CONSTRAINT pk_parcurs PRIMARY KEY(numar_traseu, cod_statie));

CREATE TABLE OBSERVATIE(
cod_observatie  NUMBER(5)      CONSTRAINT pk_observ PRIMARY KEY,
descriere      VARCHAR2(256)   NOT NULL,
data_obs        DATE           NOT NULL,
cod_vehicul     NUMBER(5)      NOT NULL,
CONSTRAINT fk_observ_veh FOREIGN KEY (cod_vehicul) REFERENCES
VEHICUL(cod_vehicul));

CREATE TABLE ANGAJAT(
cod_angajat     NUMBER(5)      CONSTRAINT pk_angajat PRIMARY KEY,
nume            VARCHAR2(25)   NOT NULL,
prenume         VARCHAR2(25)   NOT NULL,
UNIQUE,
numar_telefon  CHAR(12)       UNIQUE,
data_angajare   DATE          DEFAULT TO_DATE('01-01-2000', 'DD-MM-
YYYY'),
salariu         NUMBER(5)      NOT NULL CHECK(salariu > 0),
tip_job         VARCHAR2(25)   NOT NULL,
cod_adresa      NUMBER(5),
CONSTRAINT fk_ang_adr REFERENCES
ADRESA(cod_adresa) ON DELETE SET NULL,
cod_depou       NUMBER(5)      CONSTRAINT fk_ang_dep REFERENCES
DEPOU(cod_depou) ON DELETE SET NULL);

CREATE TABLE SOFER(
cod_sofer        NUMBER(5)      CONSTRAINT pk_sofer PRIMARY KEY,
experienta      NUMBER(2)      NOT NULL,
categorie_permis VARCHAR2(5)   NOT NULL CHECK(categorie_permis IN('BUS',
'TRAM', 'TRL')),
data_expirare   DATE          DEFAULT TO_DATE('31/05/2025', 'dd-mm-
YYYY'),
CONSTRAINT fk_sofer_ang FOREIGN KEY(cod_sofer) REFERENCES ANGAJAT(cod_angajat) ON
DELETE CASCADE);

CREATE TABLE MECANIC(

```

```

cod_mecanic          NUMBER(5)           CONSTRAINT pk_mecanic PRIMARY KEY,
specializare         VARCHAR2(20),
CONSTRAINT fk_mec_ang FOREIGN KEY(cod_mecanic) REFERENCES ANGAJAT(cod_angajat) ON
DELETE CASCADE);

CREATE TABLE INGINER(
cod_inginer          NUMBER(5)           CONSTRAINT pk_inginer PRIMARY KEY,
institutie_absolvita VARCHAR2(100)      NOT NULL,
CONSTRAINT fk_ing_ang FOREIGN KEY(cod_inginer) REFERENCES ANGAJAT(cod_angajat) ON
DELETE CASCADE);

CREATE TABLE CURSA(
cod_cursa            NUMBER(8)           CONSTRAINT pk_cursa PRIMARY KEY,
cod_sofer             NUMBER(5)           CONSTRAINT fk_cursa_sof REFERENCES
SOFER(cod_sofer) ON DELETE CASCADE,
cod_vehicul_tr        NUMBER(5)           CONSTRAINT fk_cursa_veh REFERENCES
VEHICUL_TRANSPORT(cod_vehicul_tr) ON DELETE CASCADE,
numar_traseu          NUMBER(3)           CONSTRAINT fk_cursa_trs REFERENCES
TRASEU(numar_traseu) ON DELETE CASCADE,
data_cursa            DATE                NOT NULL,
ora_preluare          DATE                NOT NULL,
ora_incepere_ture    DATE                NOT NULL,
ora_finalizare_ture  DATE                NOT NULL,
ora_predare           DATE                NOT NULL,
numar_ture             NUMBER(3)           NOT NULL,
CONSTRAINT sof_zi_unic UNIQUE(cod_sofer, data_cursa),
CONSTRAINT verif_ore CHECK(ora_incepere_ture - ora_preluare > 0
                        AND ora_predare - ora_finalizare_ture > 0));

CREATE TABLE PARTICIPARE_INTERVENTIE(
cod_participare       NUMBER(5)           CONSTRAINT pk_part PRIMARY KEY,
cod_mecanic            NUMBER(5)           CONSTRAINT fk_part_mec REFERENCES
MECANIC(cod_mecanic) ON DELETE CASCADE,
cod_vehicul_int        NUMBER(5)           CONSTRAINT fk_part_veh REFERENCES
VEHICUL_INTERVENTIE(cod_vehicul_int) ON DELETE CASCADE,
numar_interventie     NUMBER(5)           CONSTRAINT fk_part_int REFERENCES
INTERVENTIE(numar_interventie) ON DELETE CASCADE);

CREATE TABLE ATELIER(
cod_atelier            NUMBER(5)           CONSTRAINT pk_atelier PRIMARY KEY,
denumire               VARCHAR2(50)        UNIQUE NOT NULL,
an_infiintare          DATE                NOT NULL,
capacitate             NUMBER(3)           CHECK(capacitate >= 1),
cod_adresa              NUMBER(5)
CONSTRAINT fk_atelier REFERENCES ADRESA(cod_adresa) ON DELETE SET NULL,
CONSTRAINT addr_unic_atel UNIQUE(cod_adresa));

CREATE TABLE EVENIMENT_CIRCULATIE(
cod_eveniment          NUMBER(5)           CONSTRAINT pk_event PRIMARY KEY,
data_producere         DATE                NOT NULL,
localizare              VARCHAR2(100),
descriere              VARCHAR2(100),
numar_interventie      NUMBER(5)
CONSTRAINT fk_event_int REFERENCES INTERVENTIE(numar_interventie) ON DELETE SET
NULL,
CONSTRAINT nr_int_unic UNIQUE(numar_interventie));

CREATE TABLE IMPLICA(
cod_eveniment          NUMBER(5)           CONSTRAINT fk_imp_event REFERENCES
EVENIMENT_CIRCULATIE(cod_eveniment) ON DELETE CASCADE,
cod_vehicul_tr          NUMBER(5)           CONSTRAINT fk_imp_vehic REFERENCES
VEHICUL_TRANSPORT(cod_vehicul_tr) ON DELETE CASCADE,
daune                  VARCHAR2(256)        NOT NULL,
CONSTRAINT pk_imp PRIMARY KEY (cod_eveniment, cod_vehicul_tr));

CREATE TABLE REPARATIE(
cod_reparatie          NUMBER(5)           CONSTRAINT pk_rep PRIMARY KEY,
data_incepere          DATE                NOT NULL,

```

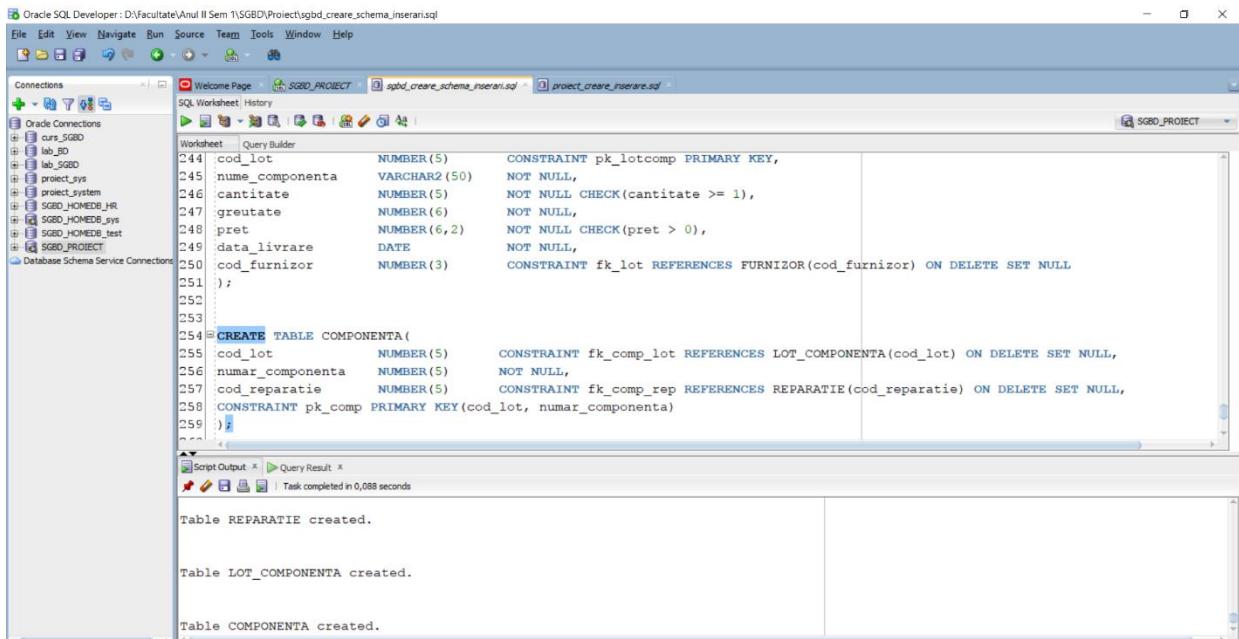
```

data_finalizare      DATE,
cod_vehicul_tr      NUMBER(5)      CONSTRAINT fk_rep_veh REFERENCES
VEHICUL_TRANSPORT(cod_vehicul_tr) ON DELETE SET NULL,
cod_inginer          NUMBER(5)      CONSTRAINT fk_rep_ing REFERENCES
INGINER(cod_inginer) ON DELETE SET NULL,
cod_atelier          NUMBER(5)      CONSTRAINT fk_rep_atl REFERENCES
ATELIER(cod_atelier) ON DELETE SET NULL);

CREATE TABLE LOT_COMPONENTA(
cod_lot              NUMBER(5)      CONSTRAINT pk_lotcomp PRIMARY KEY,
nume_componenta     VARCHAR2(50)   NOT NULL,
cantitate            NUMBER(5)      NOT NULL CHECK(cantitate >= 1),
greutate             NUMBER(6)      NOT NULL,
pret                 NUMBER(6,2)    NOT NULL CHECK(pret > 0),
data_livrare         DATE          NOT NULL,
cod_furnizor        NUMBER(3)      CONSTRAINT fk_lot REFERENCES
FURNIZOR(cod_furnizor) ON DELETE SET NULL);

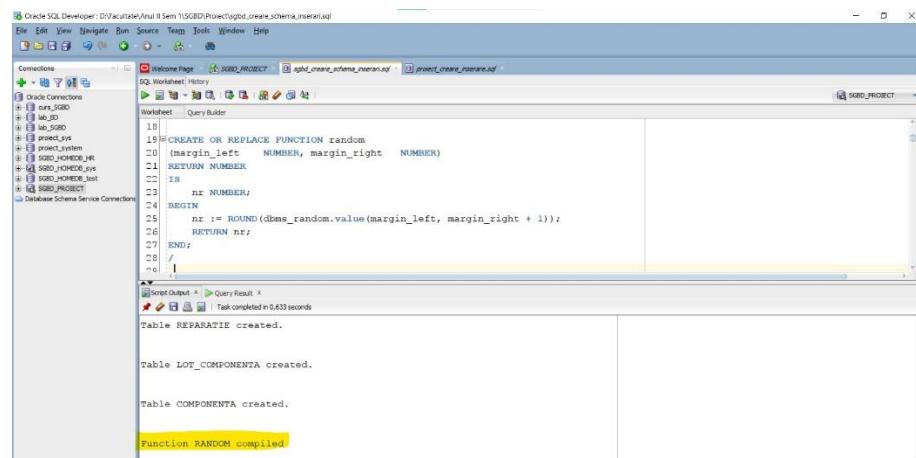
CREATE TABLE COMPONENTA(
cod_lot              NUMBER(5)      CONSTRAINT fk_comp_lot REFERENCES
LOT_COMPONENTA(cod_lot) ON DELETE SET NULL,
numar_componenta    NUMBER(5)      NOT NULL,
cod_reparatie        NUMBER(5)      CONSTRAINT fk_comp_rep REFERENCES
REPARATIE(cod_reparatie) ON DELETE SET NULL,
CONSTRAINT pk_comp PRIMARY KEY(cod_lot, numar_componenta));

```



Cerință 5: Inserarea de informații în tabele

```
--functie care imi genereaza un numar aleator intr-un interval
--al carui capete sunt date ca parametri de intrare
CREATE OR REPLACE FUNCTION random
(margin_left      NUMBER, margin_right      NUMBER)
RETURN NUMBER
IS
    nr NUMBER;
BEGIN
    nr := TRUNC(dbms_random.value(margin_left, margin_right + 1));
    RETURN nr;
END;
/
```



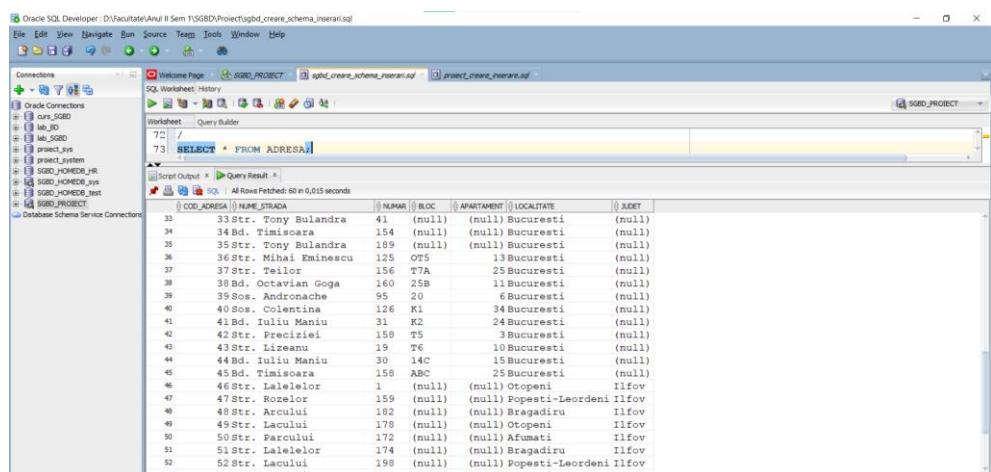
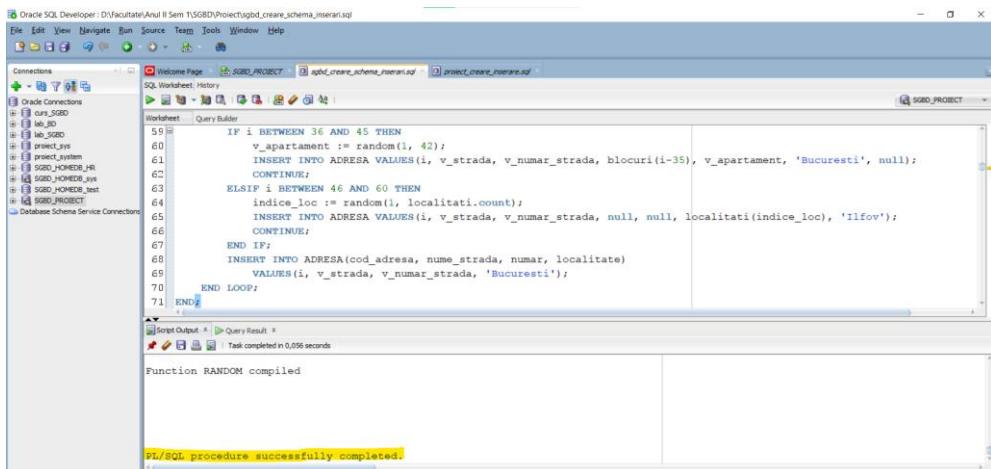
```
CREATE TYPE t_denumiri IS TABLE OF VARCHAR2(50);
/


DECLARE
    v_strazi    t_denumiri := t_denumiri('Str. Mihai Eminescu', 'Calea Dorobantilor', 'Bd. Unirii', 'Str. Lizeanu', 'Bd. Bucurestii Noi', 'Bd. Timisoara', 'Bd. Theodor Pallady', 'Str. Tony Bulandra', 'Sos. Colentina', 'Str. Preciziei', 'Bd. Octavian Goga', 'Str. Nitu Vasile', 'Str. Gioachino Rossini', 'Str. Teilor', 'Bd. Iuliu Maniu', 'Bd. Dacia', 'Sos. Andronache', 'Str. Lalelelor', 'Str. Parcului', 'Str. Arcului', 'Str. Eroilor', 'Str. Rozelor', 'Str. Lacului');
    localitati t_denumiri := t_denumiri('Otopeni', 'Voluntari', 'Bragadiru', 'Afumati', 'Popesti-Leordeni');
    blocuri    t_denumiri := t_denumiri('OT5', 'T7A', '25B', '20', 'K1', 'K2', 'T5', 'T6', '14C', 'ABC');
    v_strada    VARCHAR2(50);
    indice_strada NUMBER;
    indice_loc    NUMBER;
    v_numar_strada NUMBER;
    v_apartament NUMBER;
BEGIN
    FOR i IN 1..60 LOOP
        IF i BETWEEN 1 AND 45 THEN
            indice_strada := random(1, 17);
        ELSE
            indice_strada := random(18, 23);
        END IF;
        v_strada := v_strazi(indice_strada);
        v_numar_strada := random(1, 200);
```

```

        IF i BETWEEN 36 AND 45 THEN
            v_apartament := random(1, 42);
            INSERT INTO ADRESA VALUES(i, v_strada, v_numar_strada, blocuri(i-35),
v_apartament, 'Bucuresti', null);
            CONTINUE;
        ELSIF i BETWEEN 46 AND 60 THEN
            indice_loc := random(1, localitati.count);
            INSERT INTO ADRESA VALUES(i, v_strada, v_numar_strada, null, null,
localitati(indice_loc), 'Ilfov');
            CONTINUE;
        END IF;
        INSERT INTO ADRESA(cod_adresa, nume_strada, numar, localitate)
            VALUES(i, v_strada, v_numar_strada, 'Bucuresti');
    END LOOP;
END;
/

```



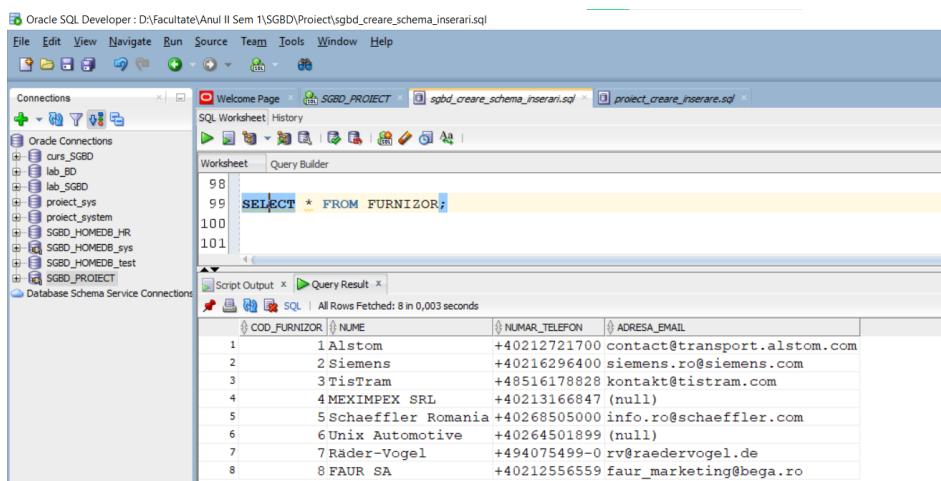
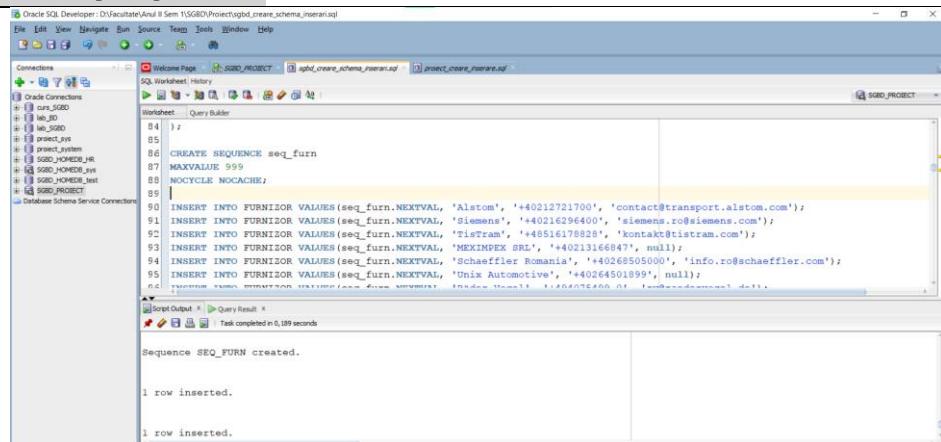
```

CREATE SEQUENCE seq_furn
MAXVALUE 999
NOCYCLE NOCACHE;

INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'Alstom', '+40212721700',
'contact@transport.alstom.com');
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'Siemens', '+40216296400',
'siemens.ro@siemens.com');
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'TisTram', '+48516178828',
'kontakt@tistram.com');

```

```
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'MEXIMPEX SRL', '+40213166847',
null);
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'Schaeffler Romania',
'+40268505000', 'info.ro@schaeffler.com');
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'Unix Automotive', '+40264501899',
null);
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'Räder-Vogel', '+494075499-0',
'rv@raedervogel.de');
INSERT INTO FURNIZOR VALUES(seq_furn.NEXTVAL, 'FAUR SA', '+40212556559',
'faur_marketing@bega.ro');
```



```
ALTER TABLE MODEL MODIFY denumire VARCHAR2(64);
```

```
CREATE SEQUENCE seq_model
INCREMENT BY 1 START WITH 1000
MAXVALUE 99999
NOCYCLE NOCACHE;

INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Kent C12', 'autobuz', 9, 28, 65,
80.6);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Kent C18', 'autobuz', 15, 52, 160,
75.8);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Trollino 12', 'troleibuz', 11, 32,
74, 75);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, '415T', 'troleibuz', 8, 28, 88, 55);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Citaro (Euro III)', 'autobuz', 10,
40, 81, 65.5);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Citaro 2 (Euro IV)', 'autobuz', 10,
40, 81, 69.8);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Imperio Metropolitan', 'tramvai',
32, 80, 320, 66.6);
```

```

INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'T4R', 'tramvai', 26, 48, 256, 55.6);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, '260', 'autobuz', 8, 32, 88, 52);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'Citelis 12T', 'troleibuz', 16, 50, 160, 77.8);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'V3A-93', 'tramvai', 24, 66, 250, 50.6);
INSERT INTO MODEL VALUES(seq_model.NEXTVAL, 'm5.65', 'tramvai', 14, 32, 74, 80.6);
--pentru vehiculele de interventie (tramvai plug)
INSERT INTO MODEL(cod_model, denumire) VALUES(seq_model.NEXTVAL, '1VU');

```

COD_MODEL	DENUMIRE	TIPO_MECANO_TRANSPORT	LUNGIME	NUMAR_SEAUNE	CAPACITATE_CULATOARE	VITEZA_MAXIMA
2	1001 Kenvi C18	autobuz	15	52	160	75.8
3	1002 Trollino 12	troleibuz	11	32	74	75
4	1003 415T	troleibuz	8	28	88	55
5	1004 Citaro (Euro III)	autobuz	10	40	81	65.5
6	1005 Citaro 2 (Euro IV)	autobuz	10	40	81	69.8
7	1006 Imperio Metropolitan tramvai	tramvai	32	80	320	66.6
8	1007 T4R	tramvai	26	48	256	55.6
9	1008 260	autobuz	8	32	88	52
10	1009 Citelis 12T	troleibuz	16	50	160	77.8
11	1010 V3A-93	tramvai	24	66	250	50.6
12	1011 m5.65	tramvai	14	32	74	80.6
13	1012 1VU	(null)	(null)	(null)	(null)	(null)

```

ALTER TABLE TRASEU MODIFY data_infiintare DEFAULT TO_DATE('01/09/1990',
'DD/MM/YYYY');

```

```

INSERT INTO TRASEU VALUES(1, 35.7, 'urban', INTERVAL '8' MINUTE, 35000, DEFAULT,
null);
INSERT INTO TRASEU VALUES(5, 12, 'urban', INTERVAL '15' MINUTE, 5000, DEFAULT,
TO_DATE('01/12/2023', 'DD/MM/YYYY'));
INSERT INTO TRASEU VALUES(14, 8.9, 'urban', INTERVAL '20' MINUTE, 1500,
TO_DATE('01/05/2001', 'DD/MM/YYYY'), null);
INSERT INTO TRASEU VALUES(21, 33.2, 'urban', INTERVAL '5' MINUTE, 22000, DEFAULT,
null);
INSERT INTO TRASEU VALUES(66, 8.2, 'urban', INTERVAL '20' MINUTE, 4500, DEFAULT,
null);
INSERT INTO TRASEU VALUES(73, 18, 'urban', INTERVAL '12' MINUTE, 8200,
TO_DATE('15/12/2024', 'DD/MM/YYYY'), null);
INSERT INTO TRASEU VALUES(85, 12, 'urban', INTERVAL '9' MINUTE, 5100, DEFAULT,
null);
INSERT INTO TRASEU VALUES(182, 14.5, 'urban', INTERVAL '13' MINUTE, 7600,
DEFAULT, null);
INSERT INTO TRASEU VALUES(104, 21, 'urban', INTERVAL '4' MINUTE, 13200,
TO_DATE('01/05/2001', 'DD/MM/YYYY'), null);
INSERT INTO TRASEU VALUES(381, 15, 'urban', INTERVAL '7' MINUTE, 14400,
TO_DATE('01/05/2001', 'DD/MM/YYYY'), null);
INSERT INTO TRASEU VALUES(605, 12, 'urban', INTERVAL '20' MINUTE, 4000,
TO_DATE('01/12/2023', 'DD/MM/YYYY'), null);
INSERT INTO TRASEU VALUES(441, 42, 'regional', INTERVAL '40' MINUTE, 3500,
DEFAULT, null);
INSERT INTO TRASEU VALUES(413, 28, 'regional', INTERVAL '50' MINUTE, 500,
DEFAULT, null);
INSERT INTO TRASEU VALUES(901, 13, 'de noapte', INTERVAL '30' MINUTE, 2500,
TO_DATE('01/06/2024', 'DD/MM/YYYY'), null);
INSERT INTO TRASEU VALUES(902, 7.9, 'de noapte', INTERVAL '60' MINUTE, 1500,
TO_DATE('01/06/2024', 'DD/MM/YYYY'), null);

--SELECT * FROM TRASEU;

```

The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, there is a code block for inserting data into the TRASEU table. In the Script Output tab, the results of the query are displayed in a table with columns: NUMAR_TRASEU, LUNGIME, CATEGORIE, FRECVENTA, MEDIE_CALATORI, DATA_INFINTARE, and DATA_SUSPENDARE. The data consists of 15 rows of travel statistics.

NUMAR_TRASEU	LUNGIME	CATEGORIE	FRECVENTA	MEDIE_CALATORI	DATA_INFINTARE	DATA_SUSPENDARE
1	35.7	urban	+00 00:08:00.000000	35000	01-SEP-90	(null)
2	5	12 urban	+00 00:15:00.000000	5000	01-SEP-90	01-DEC-23
3	14	8.9 urban	+00 00:20:00.000000	1500	01-MAY-01	(null)
4	21	33.2 urban	+00 00:05:00.000000	22000	01-SEP-90	(null)
5	66	8.2 urban	+00 00:20:00.000000	4500	01-SEP-90	(null)
6	73	18 urban	+00 00:12:00.000000	8200	15-DEC-24	(null)
7	85	12 urban	+00 00:09:00.000000	5100	01-SEP-90	(null)
8	182	14.5 urban	+00 00:13:00.000000	7600	01-SEP-90	(null)
9	104	21 urban	+00 00:04:00.000000	13200	01-MAY-01	(null)
10	381	15 urban	+00 00:07:00.000000	14400	01-MAY-01	(null)
11	605	12 urban	+00 00:20:00.000000	4000	01-SEP-23	(null)
12	441	42 regional	+00 00:40:00.000000	3500	01-SEP-90	(null)
13	413	28 regional	+00 00:50:00.000000	500	01-SEP-90	(null)
14	901	13 de noapte	+00 00:30:00.000000	2500	01-JUN-24	(null)
15	902	7.9 de noapte	+00 00:00.000000	1500	01-JUN-24	(null)

```

ALTER TABLE STATIE MODIFY nume VARCHAR2(64);

DECLARE
    v_statii_1    t_denumiri := t_denumiri('Lizeanu', 'Bucur Obor', 'Doamna Ghica', 'Bd. Ferdinand', 'Piata Iancului', 'Vasile Lascăr', 'Vatra Luminoasă', 'Pasaj Victoria', 'Piata Muncii', 'Stadionul Național', 'Baba Novac', 'I.O.R.', 'Piata Titan', 'Piata Dorobanti', 'Arcul de Triumf', 'Piata Charles de Gaulle', 'Parcul Herastrau', 'Piata 1 Mai', 'Clăbucet', 'Bd. Laminorului', 'Parc Bazilescu', 'Circul Globus', 'Piata Gemeni', 'Armeneasca', 'Scoala Iancului', 'Piata Sudului', 'Piata Crangasi', 'Gradina Cismigiu', 'Gradina Botanica', 'Baicului', 'Raul Doamnei', 'Valea Oltului', 'Valea Ialomitei', 'Parc Drumul Taberei', 'Favorit', 'Orizont', 'Izvor', 'Piata Unirii 1', 'Piata Unirii 2', 'Tineretului', 'Timpuri Noi', 'Piata Resita', 'Piata Sf. Gheorghe', 'Pasaj Colentina', 'Cimitirul Bellu', 'Piata Danny Huwe', 'Piata Regina Maria', 'Gara de Nord', 'Gara Basarab', 'Podul Grant', 'Lujerului', 'Pacii', 'Dristorului', 'Universitate', 'Piata Rosetti', 'Mihai Eminescu', 'Valea Cascadelor', 'Preciziei', 'C.F.R. Progresul', 'I.M.G.B.', 'Delfinului', 'Granitul', 'Sfanta Vineri');

    strazi_1    t_denumiri := t_denumiri('Sos. Stefan Cel Mare', 'Sos. Colentina', 'Bd. Nicolae Grigorescu', 'Bd. Regina Elisabeta', 'Str. Barbu Vacarescu', 'Calea Floreasca', 'Str. Turnu Magurele', 'Sos. Giurgiului', 'Sos. Alexandriei', 'Calea Serban Voda', 'Bd. Carol I', 'Bd. Drumul Taberei', 'Sos. Virtutii', 'Sos. Pantelimon', 'Bd. Iancu de Hunedoara', 'Bd. Ion Mihalache');

    v_statii_2  t_denumiri := t_denumiri('Gladiolelor', 'Biserica', 'Acvila', 'Crinului', 'Cimitir Ghencea', 'Pasaj Lujerului', 'Crizantemelor', 'Primaria Clinceni', 'Soseaua de Centura', 'Cartierul Latin');
    strazi_2    t_denumiri := t_denumiri('Str. Lalelelor', 'Str. Parcului', 'Str. Arcului', 'Str. Eroilor');
    v_localitate  VARCHAR2(25);
    i_strada      NUMBER;

BEGIN
    --statiile din Bucuresti
    FOR i IN 1..v_statii_1.last LOOP
        i_strada := random(1, strazi_1.count);
        INSERT INTO STATIE VALUES(1000 + i, v_statii_1(i), 'Bucuresti', strazi_1(i_strada));
    END LOOP;
    --statiile din afara Bucurestului
    FOR i IN 1..v_statii_2.last LOOP
        i_strada := random(1, strazi_2.count);
        IF i MOD 3 = 0 THEN v_localitate := 'Bragadiru';
        ELSIF i MOD 3 = 1 THEN v_localitate := 'Domnesti';
        ELSE v_localitate := 'Clinceni';
    END LOOP;
END;

```

```

        END IF;
        INSERT INTO STATIE VALUES(2000 + i, v_statii_2(i), v_localitate,
strazi_2(i_strada));
    END LOOP;
END;
/
SELECT * FROM STATIE;

```

The screenshot shows the Oracle SQL Developer interface with a script named 'spbd_create_schema_inserari.sql' running. The output pane displays the results of the 'SELECT * FROM STATIE;' query, which shows 66 rows of data inserted into the STATIE table.

COD_STATIE	NUME	LOCALITATE	ARTERA
50	1050 Podul Grant	Bucuresti Bd. Drumul Taberei	
51	1051 Lujerului	Bucuresti Bd. Regina Elisabeta	
52	1052 Pacii	Bucuresti Bd. Drumul Taberei	
53	1053 Dristorului	Bucuresti Sos. Pantelimon	
54	1054 Universitate	Bucuresti Bd. Regina Elisabeta	
55	1055 Piata Rosetti	Bucuresti Sos. Giurgiului	
56	1056 Mai Eminescu	Bucuresti Bd. Iancu de Hunedoara	
57	1057 Strada Cascadelor	Bucuresti Bd. Pantelimon	
58	1058 Precisel	Bucuresti Bd. Colol I	
59	1059 C.R. Progresul	Bucuresti Sos. Colentina	
60	1060 I.M.G.B.	Bucuresti Galata Floresasca	
61	1061 Delfinului	Bucuresti Bd. Ion Mihalache	
62	1062 Granitul	Bucuresti Bd. Ion Mihalache	
63	1063 Sfanta Vineri	Bucuresti Sos. Virgilului	
64	2001 Gladiolelor	Domenesti Str. Eroilor	
65	2002 Biserica	Clinceni Str. Arcului	
66	2003 Acvila	Bragadiru Str. Lalelelor	

```

DECLARE
    cauze_interventii t_denumiri := t_denumiri('Copaci cazuti pe reteaua de
contact', 'Eveniment circulatie',
'Lipsa tensiune', 'Deszapezire', 'Eveniment circulatie');
    v_data      TIMESTAMP WITH TIME ZONE;
    v_durata    NUMBER;
    indice      NUMBER;
BEGIN
    FOR i IN 800..830 LOOP
        v_data := SYSDATE - INTERVAL '1' DAY * random(0, 365) + INTERVAL '1'
SECOND * random(0, 86400);
        v_durata := random(1, 30) * 10;
        indice := i MOD 5 + 1;
        INSERT INTO INTERVENTIE VALUES(i, v_data, cauze_interventii(indice),
TO_DSINTERVAL('PT' || v_durata || 'M'));
    END LOOP;
END;
/
SELECT * FROM INTERVENTIE;

```

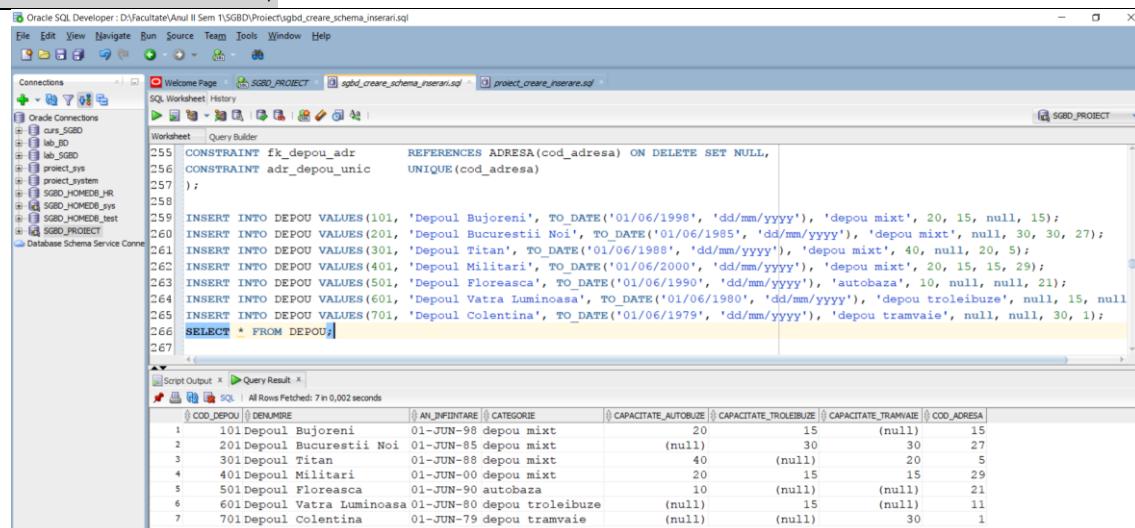
The screenshot shows the Oracle SQL Developer interface with a script named 'spbd_create_schema_inserari.sql' running. The output pane displays the results of the 'SELECT * FROM INTERVENTIE;' query, which shows 20 rows of data inserted into the INTERVENTIE table.

NUMAR_INTERVENTIE	DATA_INTERVENTIE	CAUZA	DURATA
8	807.29.-NOV-24 05.19.20.000000000 PM EUROPE/BUCHAREST Lipsa tensiune	+00 00:21:00.0000000	
9	808.30.-JUL-24 02.20.41.000000000 AM EUROPE/BUCHAREST Deszapezire	+00 00:20:00.0000000	
10	809.18.-JAN-24 04.57.40.000000000 PM EUROPE/BUCHAREST Eveniment circulatie	+00 03:20:00.0000000	
11	810.11.-SEP-24 12.27.44.000000000 PM EUROPE/BUCHAREST Copaci cazuti pe reteaua de contact	+00 02:00:00.0000000	
12	811.21.-DEC-24 09.47.42.000000000 PM EUROPE/BUCHAREST Eveniment circulatie	+00 04:40:00.0000000	
13	812.04.-AUG-24 07.38.03.000000000 PM EUROPE/BUCHAREST Lipsa tensiune	+00 00:50:00.0000000	
14	813.03.-JAN-25 07.00.35.000000000 PM EUROPE/BUCHAREST Deszapezire	+00 02:30:00.0000000	
15	814.23.-NOV-24 07.07.39.000000000 PM EUROPE/BUCHAREST Eveniment circulatie	+00 03:40:00.0000000	
16	815.15.-JUL-24 05.21.23.000000000 AM EUROPE/BUCHAREST Copaci cazuti pe reteaua de contact	+00 02:30:00.0000000	
17	816.21.-MAY-24 09.24.13.000000000 AM EUROPE/BUCHAREST Eveniment circulatie	+00 00:20:00.0000000	
18	817.15.-JAN-24 05.35.02.000000000 AM EUROPE/BUCHAREST Lipsa tensiune	+00 02:30:00.0000000	
19	818.18.-MAY-24 12.32.22.000000000 AM EUROPE/BUCHAREST Deszapezire	+00 03:30:00.0000000	
20	819.25.-DEC-24 08.51.35.000000000 AM EUROPE/BUCHAREST Eveniment circulatie	+00 04:20:00.0000000	

```

INSERT INTO DEPOU VALUES(101, 'Depoul Bujoreni', TO_DATE('01/06/1998',
'dd/mm/yyyy'), 'depou mixt', 20, 15, null, 15);
INSERT INTO DEPOU VALUES(201, 'Depoul Bucurestii Noi', TO_DATE('01/06/1985',
'dd/mm/yyyy'), 'depou mixt', null, 30, 30, 27);
INSERT INTO DEPOU VALUES(301, 'Depoul Titan', TO_DATE('01/06/1988',
'dd/mm/yyyy'), 'depou mixt', 40, null, 20, 5);
INSERT INTO DEPOU VALUES(401, 'Depoul Militari', TO_DATE('01/06/2000',
'dd/mm/yyyy'), 'depou mixt', 20, 15, 15, 29);
INSERT INTO DEPOU VALUES(501, 'Depoul Floreasca', TO_DATE('01/06/1990',
'dd/mm/yyyy'), 'autobaza', 10, null, null, 21);
INSERT INTO DEPOU VALUES(601, 'Depoul Vatra Luminoasa', TO_DATE('01/06/1980',
'dd/mm/yyyy'), 'depou troleibuze', null, 15, null, 11);
INSERT INTO DEPOU VALUES(701, 'Depoul Colentina', TO_DATE('01/06/1979',
'dd/mm/yyyy'), 'depou tramvaie', null, null, 30, 1);
SELECT * FROM DEPOU;

```



```

CREATE TYPE t_coduri IS TABLE OF NUMBER;
/
CREATE TYPE t_date IS TABLE OF DATE;
/

DECLARE
    v_coduri_modele t_coduri;
    producatori t_denumiri;
    an_achiz DATE;
    ani_achiz t_date;
    v_date_verif t_date := t_date(TO_DATE('15/10/2024', 'dd/mm/yyyy'),
TO_DATE('15/10/2022', 'dd/mm/yyyy'), TO_DATE('15/11/2018', 'dd/mm/yyyy'));
    nr_linie NUMBER := 0;
    v_data DATE;
    v_cod_veh NUMBER;
    nr_veh_per_dep NUMBER := 1;
    v_stare VARCHAR2(20);
    v_km NUMBER;

BEGIN
    --AUTOBUZE
    SELECT cod_model BULK COLLECT INTO v_coduri_modele FROM MODEL WHERE
tip_mijloc_transport LIKE 'autobuz';
    producatori := t_denumiri('Otokar', 'Otokar', 'Mercedes Benz', 'Mercedes Benz',
'Ikarus');
    ani_achiz := t_date(TO_DATE('2018', 'YYYY'), TO_DATE('2018', 'YYYY'),
TO_DATE('2007', 'YYYY'), TO_DATE('2009', 'YYYY'), TO_DATE('1991', 'YYYY'));
    FOR i IN 1..v_coduri_modele.last LOOP
        FOR i_dep IN (SELECT cod_depou FROM depou WHERE capacitate_autobuze IS NOT
NULL) LOOP
            FOR k IN 1..nr_veh_per_dep LOOP
                v_cod_veh := 1000 + nr_linie;

```

```

        v_km := random(10000, 500000);
        IF nr_linie MOD 4 = 0 THEN v_data := v_date_verif(1);
        ELSE v_data := TO_DATE('06/01/2025', 'dd/mm/yyyy');
        END IF;
        IF nr_linie MOD 6 = 0 THEN v_stare := 'in reparatie';
        ELSIF nr_linie MOD 8 = 0 THEN v_stare := 'defect';
        ELSE v_stare := 'functional';
        END IF;
        INSERT INTO vehicul
        VALUES(v_cod_veh, producatori(i), ani_achiz(i), v_data, v_stare,
v_km, 'transport', i_dep.cod_depou, v_coduri_modele(i));
        nr_linie := nr_linie + 1;
    END LOOP;
    nr_veh_per_dep := nr_veh_per_dep MOD 3 + 1;
END LOOP;
END LOOP;

--TROLEIBUZE
SELECT cod_model BULK COLLECT INTO v_coduri_modele FROM MODEL WHERE
tip_mijloc_transport LIKE 'troleibuz';
producatori := t_denumiri('Solaris', 'Ikarus', 'Astra Irisbus');
ani_achiz := t_date(TO_DATE('2022', 'yyyy'), TO_DATE('1999', 'yyyy'),
TO_DATE('2007', 'yyyy'));
FOR i IN 1..v_coduri_modele.last LOOP
    FOR i_dep IN (SELECT cod_depou FROM depou WHERE capacitate_troleibuze IS NOT
NULL) LOOP
        FOR k IN 1..nr_veh_per_dep LOOP
            v_cod_veh := 5000 + nr_linie;
            v_km := random(10000, 500000);

            IF nr_linie MOD 5 = 0 THEN v_data := v_date_verif(2);
            ELSE v_data := TO_DATE('06/01/2025', 'dd/mm/yyyy');
            END IF;

            IF nr_linie MOD 8 = 0 THEN v_stare := 'in reparatie';
            ELSIF nr_linie MOD 5 = 0 THEN v_stare := 'defect';
            ELSE v_stare := 'functional';
            END IF;

            INSERT INTO vehicul
            VALUES(v_cod_veh, producatori(i), ani_achiz(i), v_data, v_stare,
v_km, 'transport', i_dep.cod_depou, v_coduri_modele(i));
            nr_linie := nr_linie + 1;
        END LOOP;
        nr_veh_per_dep := nr_veh_per_dep MOD 3 + 1;
    END LOOP;
END LOOP;

--TRAMVAIE
SELECT cod_model BULK COLLECT INTO v_coduri_modele FROM MODEL WHERE
tip_mijloc_transport LIKE 'tramvai';
producatori := t_denumiri('Astra Arad', 'CKD Praga', 'R.A.T.B. URAC', 'Rathgeber
München');
ani_achiz := t_date(TO_DATE('2024', 'yyyy'), TO_DATE('1975', 'yyyy'),
TO_DATE('1999', 'yyyy'), TO_DATE('1960', 'yyyy'));
FOR i IN 1..v_coduri_modele.last LOOP
    FOR i_dep IN (SELECT cod_depou FROM depou WHERE capacitate_tramvaie IS NOT
NULL) LOOP
        FOR k IN 1..nr_veh_per_dep LOOP
            v_cod_veh := 9000 + nr_linie;
            v_km := random(10000, 500000);

            IF i = 3 THEN an_achiz := TO_DATE(random(1991, 2008), 'yyyy');
            ELSE an_achiz := ani_achiz(i);
            END IF;

            IF nr_linie MOD 3 = 0 THEN v_data := v_date_verif(1);
            ELSE v_data := TO_DATE('06/01/2025', 'dd/mm/yyyy');
            END IF;
        END LOOP;
    END LOOP;
END LOOP;

```

```

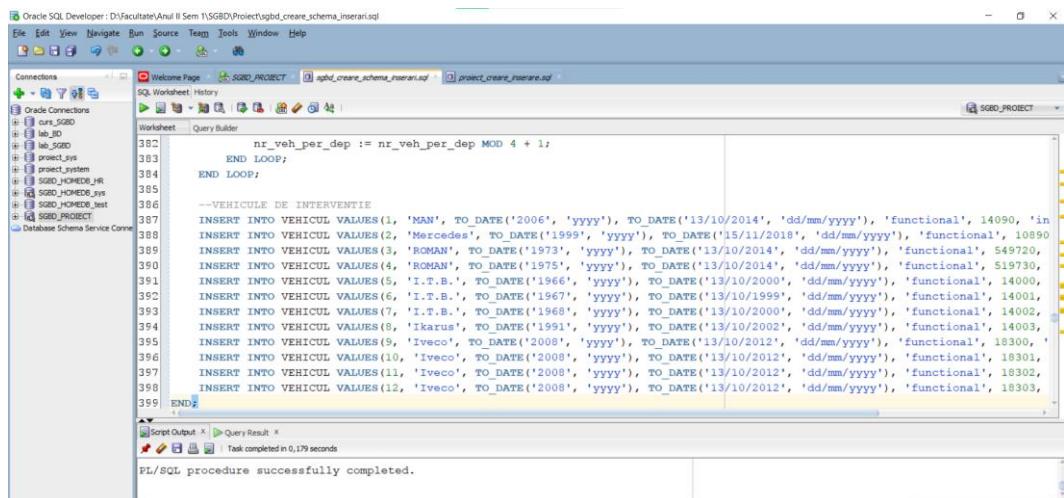
        END IF;

        IF nr_linie MOD 7 = 0 THEN v_stare := 'in reparatie';
ELSIF nr_linie MOD 6 = 0 THEN v_stare := 'defect';
ELSE v_stare := 'functional';
END IF;

        INSERT INTO vehicul
VALUES(v_cod_veh, producatori(i), an_achiz, v_data, v_stare, v_km,
'transport', i_dep.cod_depou, v_coduri_modele(i));
        nr_linie := nr_linie + 1;
    END LOOP;
    nr_veh_per_dep := nr_veh_per_dep MOD 4 + 1;
END LOOP;
END LOOP;

--VEHICULE DE INTERVENTIE
INSERT INTO VEHICUL VALUES(1, 'MAN', TO_DATE('2006', 'yyyy'),
TO_DATE('13/10/2014', 'dd/mm/yyyy'), 'functional', 14090, 'interventie', 101, null);
    INSERT INTO VEHICUL VALUES(2, 'Mercedes', TO_DATE('1999', 'yyyy'),
TO_DATE('15/11/2018', 'dd/mm/yyyy'), 'functional', 108900, 'interventie', 301,
null);
    INSERT INTO VEHICUL VALUES(3, 'ROMAN', TO_DATE('1973', 'yyyy'),
TO_DATE('13/10/2014', 'dd/mm/yyyy'), 'functional', 549720, 'interventie', 401,
null);
    INSERT INTO VEHICUL VALUES(4, 'ROMAN', TO_DATE('1975', 'yyyy'),
TO_DATE('13/10/2014', 'dd/mm/yyyy'), 'functional', 519730, 'interventie', 501,
null);
    INSERT INTO VEHICUL VALUES(5, 'I.T.B.', TO_DATE('1966', 'yyyy'),
TO_DATE('13/10/2000', 'dd/mm/yyyy'), 'functional', 14000, 'interventie', 201, 1012);
    INSERT INTO VEHICUL VALUES(6, 'I.T.B.', TO_DATE('1967', 'yyyy'),
TO_DATE('13/10/1999', 'dd/mm/yyyy'), 'functional', 14001, 'interventie', 401, 1012);
    INSERT INTO VEHICUL VALUES(7, 'I.T.B.', TO_DATE('1968', 'yyyy'),
TO_DATE('13/10/2000', 'dd/mm/yyyy'), 'functional', 14002, 'interventie', 701, 1012);
    INSERT INTO VEHICUL VALUES(8, 'Ikarus', TO_DATE('1991', 'yyyy'),
TO_DATE('13/10/2002', 'dd/mm/yyyy'), 'functional', 14003, 'interventie', 601, 1003);
    INSERT INTO VEHICUL VALUES(9, 'Iveco', TO_DATE('2008', 'yyyy'),
TO_DATE('13/10/2012', 'dd/mm/yyyy'), 'functional', 18300, 'interventie', 101, null);
    INSERT INTO VEHICUL VALUES(10, 'Iveco', TO_DATE('2008', 'yyyy'),
TO_DATE('13/10/2012', 'dd/mm/yyyy'), 'functional', 18301, 'interventie', 301, null);
    INSERT INTO VEHICUL VALUES(11, 'Iveco', TO_DATE('2008', 'yyyy'),
TO_DATE('13/10/2012', 'dd/mm/yyyy'), 'functional', 18302, 'interventie', 401, null);
    INSERT INTO VEHICUL VALUES(12, 'Iveco', TO_DATE('2008', 'yyyy'),
TO_DATE('13/10/2012', 'dd/mm/yyyy'), 'functional', 18303, 'interventie', 501, null);
END;
/

```



COD_VEHICUL_TR	PRODUCATOR	AN_ACHIZITIE	DATA_JURIFICAIRE	STARE	KILOMETRAG	INTREBUIINTARE	COD_DEP	COD_MODEL
36	1035Karus	01-JAN-91	06-JAN-25	functional	229325	transport	301	1008
37	1036Karus	01-JAN-91	15-OCT-24	in reparatie	242819	transport	401	1008
38	1037Karus	01-JAN-91	06-JAN-25	functional	80299	transport	501	1008
39	1038Karus	01-JAN-91	06-JAN-25	functional	480763	transport	501	1008
40	5039Solaris	01-JAN-22	06-JAN-25	functional	90356	transport	101	1002
41	5040Solaris	01-JAN-22	15-OCT-22	in reparatie	254086	transport	101	1002
42	5041Solaris	01-JAN-22	06-JAN-25	functional	101847	transport	101	1002
43	5042Solaris	01-JAN-22	06-JAN-25	functional	93676	transport	201	1002
44	5043Solaris	01-JAN-22	06-JAN-25	functional	208031	transport	401	1002
45	5044Solaris	01-JAN-22	06-JAN-25	functional	169220	transport	401	1002
46	5045Solaris	01-JAN-22	15-OCT-22	defect	196085	transport	601	1002
47	5046Solaris	01-JAN-22	06-JAN-25	functional	411145	transport	601	1002
48	5047Solaris	01-JAN-22	06-JAN-25	functional	131635	transport	601	1002
49	5048Ikarus	01-JAN-99	06-JAN-25	in reparatie	287513	transport	101	1003
50	5049Ikarus	01-JAN-99	06-JAN-25	functional	353110	transport	201	1003
51	5050Ikarus	01-JAN-99	15-OCT-22	defect	263159	transport	201	1003
52	5051Ikarus	01-JAN-99	06-JAN-25	functional	315735	transport	401	1003
53	5052Ikarus	01-JAN-99	06-JAN-25	functional	408297	transport	401	1003
54	5053Ikarus	01-JAN-99	06-JAN-25	functional	192001	transport	401	1003
55	5054Ikarus	01-JAN-99	06-JAN-25	functional	432666	transport	601	1003
56	5055Ierra Triebus	01-JAN-07	15-OCT-22	defect	326589	transport	101	1009

```

DECLARE
    v_coduri_veh_tr t_coduri := t_coduri();
BEGIN
    SELECT cod_vehicul BULK COLLECT INTO v_coduri_veh_tr FROM vehicul WHERE
intrebuintare = 'transport';
    FORALL i IN 1..v_coduri_veh_tr.last
        INSERT INTO vehicul_transport VALUES (v_coduri_veh_tr(i));
END;
/
SELECT * FROM vehicul_transport;

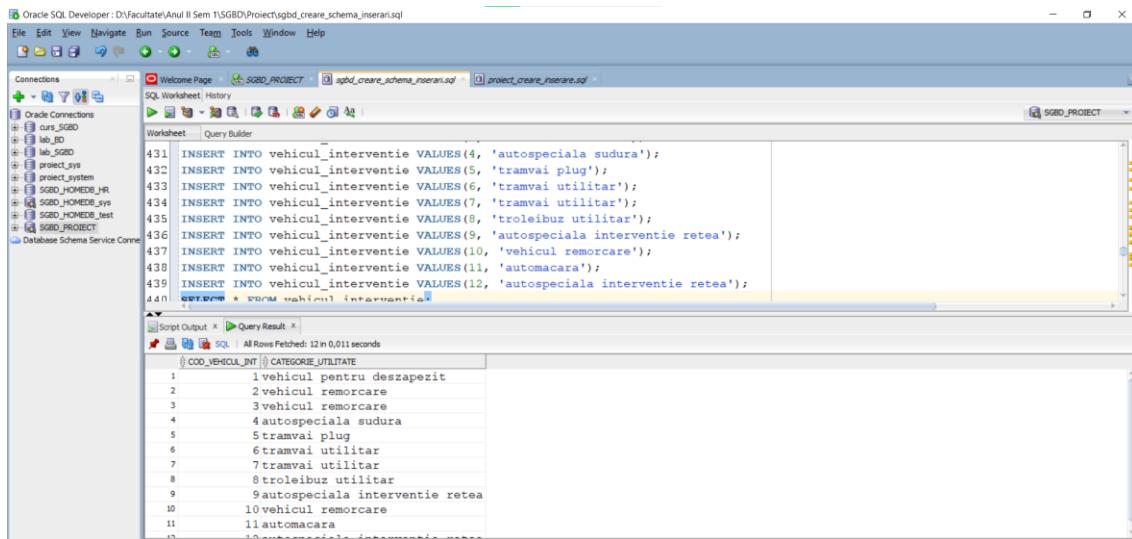
```

COD_VEHICUL_TR
1
2
3
4
5
6
7
8
9
10
1009
1010

```

INSERT INTO vehicul_interventie VALUES(1, 'vehicul pentru deszapezit');
INSERT INTO vehicul_interventie VALUES(2, 'vehicul remorcare');
INSERT INTO vehicul_interventie VALUES(3, 'vehicul remorcare');
INSERT INTO vehicul_interventie VALUES(4, 'autospeciala sudura');
INSERT INTO vehicul_interventie VALUES(5, 'tramvai plug');
INSERT INTO vehicul_interventie VALUES(6, 'tramvai utilitar');
INSERT INTO vehicul_interventie VALUES(7, 'tramvai utilitar');
INSERT INTO vehicul_interventie VALUES(8, 'troleibuz utilitar');
INSERT INTO vehicul_interventie VALUES(9, 'autospeciala interventie retea');
INSERT INTO vehicul_interventie VALUES(10, 'vehicul remorcare');
INSERT INTO vehicul_interventie VALUES(11, 'automacara');
INSERT INTO vehicul_interventie VALUES(12, 'autospeciala interventie retea');

```



```

DECLARE
    TYPE t_statii   IS TABLE OF statie%rowtype;
    TYPE t_parcurs  IS TABLE OF parcurs_traseu%rowtype;
    v_statii_buc      t_statii := t_statii();
    nr_statie_start    NUMBER;
    nr_statii_traseu   NUMBER;
    statii_5          t_parcurs := t_parcurs();
    nr_ord_statie      NUMBER;

BEGIN
    --mai intai populez cu statii traseele care nu sunt regionale
    FOR i IN (SELECT numar_traseu FROM traseu WHERE numar_traseu NOT IN(605, 441,
413)) LOOP
        nr_ord_statie := 1;
        IF i.numar_traseu MOD 2 = 0 THEN
            /*ordonez statiile dupa artera, pentru a avea o coerență a datelor
(statii consecutive pe traseu să fie localizate pe aceeași stradă) */
            SELECT * BULK COLLECT INTO v_statii_buc FROM statie WHERE localitate =
'Bucuresti' ORDER BY 4;
        ELSE
            SELECT * BULK COLLECT INTO v_statii_buc FROM statie WHERE localitate =
'Bucuresti' ORDER BY 4 DESC;
        END IF;
        nr_statie_start := random(1, 48);
        nr_statii_traseu := random(5, 15);
        FOR k IN nr_statie_start..nr_statie_start + nr_statii_traseu LOOP
            INSERT INTO PARCURS_TRASEU VALUES(i.numar_traseu,
v_statii_buc(k).cod_statie, nr_ord_statie);
            nr_ord_statie := nr_ord_statie + 1;
        END LOOP;
    END LOOP;

    --traseul 605, provizoriu, are aceleasi statii ca si traseul 5 (cel suspendat)
    SELECT * BULK COLLECT INTO statii_5 FROM parcurs_traseu WHERE numar_traseu = 5;
    FORALL i IN 1..statii_5.last
        INSERT INTO PARCURS_TRASEU VALUES(605, statii_5(i).cod_statie,
statii_5(i).numar_ordine);

    --traseele regionale au o statie in Bucuresti (aceeasi statie terminus) si
restul in localitati din judetul Ilfov
    INSERT INTO PARCURS_TRASEU VALUES(413, 1051, 1);
    INSERT INTO PARCURS_TRASEU VALUES(441, 1051, 1);

    SELECT * BULK COLLECT INTO v_statii_buc FROM statie WHERE localitate <>
'Bucuresti';
    nr_statie_start := random(1, 5);
    nr_statii_traseu := random(3, 5);
    nr_ord_statie := 2;

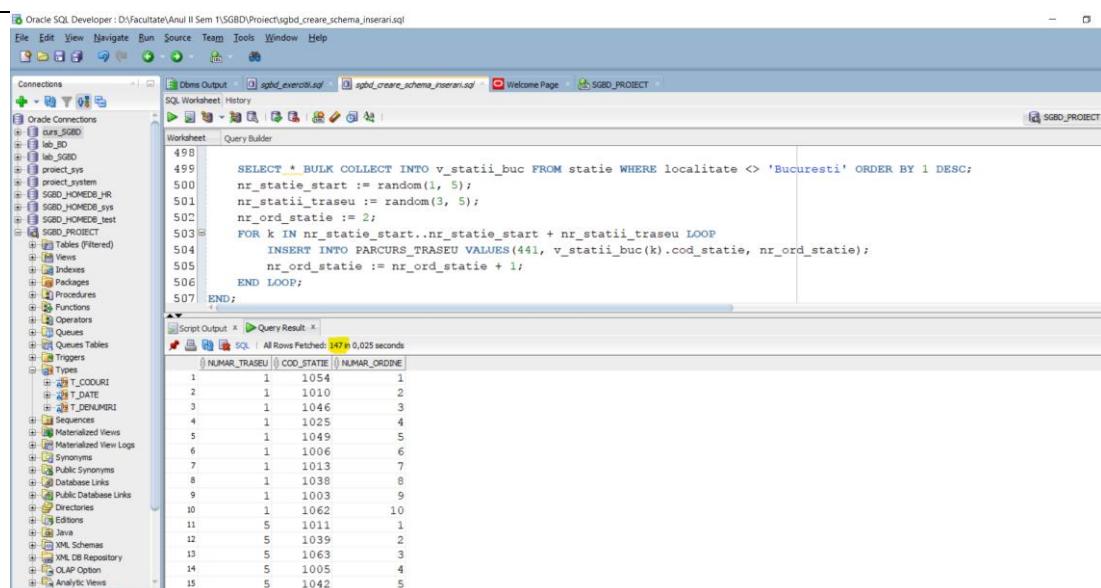
```

```

FOR k IN nr_statie_start..nr_statie_start + nr_statii_traseu LOOP
    INSERT INTO PARCURS_TRASEU VALUES(413, v_statii_buc(k).cod_statie,
nr_ord_statie);
    nr_ord_statie := nr_ord_statie + 1;
END LOOP;

SELECT * BULK COLLECT INTO v_statii_buc FROM statie WHERE localitate <>
'Bucuresti' ORDER BY 1 DESC;
nr_statie_start := random(1, 5);
nr_statii_traseu := random(3, 5);
nr_ord_statie := 2;
FOR k IN nr_statie_start..nr_statie_start + nr_statii_traseu LOOP
    INSERT INTO PARCURS_TRASEU VALUES(441, v_statii_buc(k).cod_statie,
nr_ord_statie);
    nr_ord_statie := nr_ord_statie + 1;
END LOOP;
END;
/

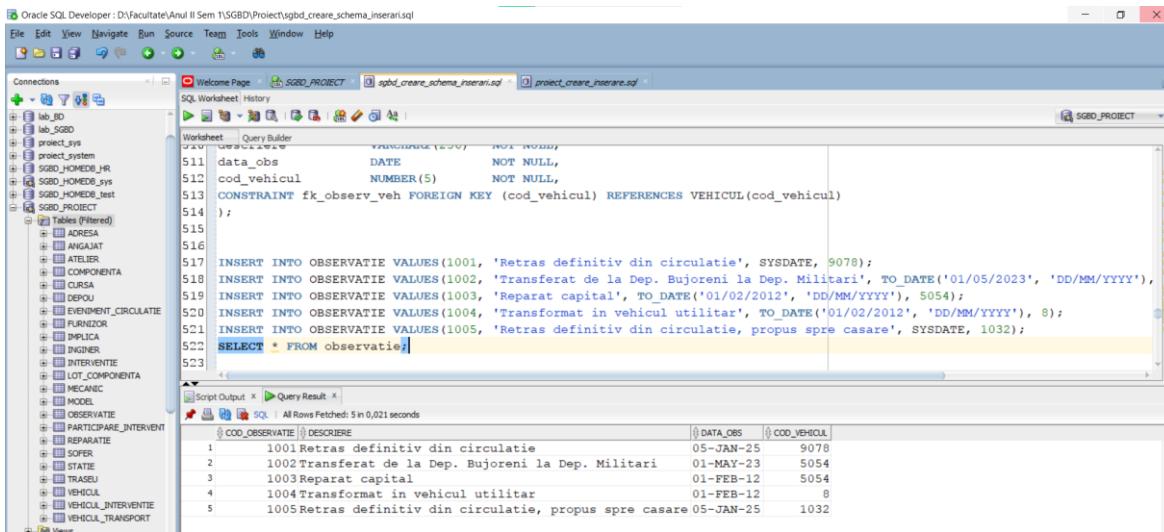
```



```

INSERT INTO OBSERVATIE VALUES(1001, 'Retras definitiv din circulatie', SYSDATE,
9078);
INSERT INTO OBSERVATIE VALUES(1002, 'Transferat de la Dep. Bujoreni la Dep.
Militari', TO_DATE('01/05/2023', 'DD/MM/YYYY'), 5054);
INSERT INTO OBSERVATIE VALUES(1003, 'Reparat capital', TO_DATE('01/02/2012',
'DD/MM/YYYY'), 5054);
INSERT INTO OBSERVATIE VALUES(1004, 'Transformat in vehicul utilitar',
TO_DATE('01/02/2012', 'DD/MM/YYYY'), 8);
INSERT INTO OBSERVATIE VALUES(1005, 'Retras definitiv din circulatie, propus spre
casare', SYSDATE, 1032);

```



```

DECLARE
    v_nume t_denumiri := t_denumiri('Ionescu', 'Popescu', 'Andrei', 'Grigore',
    'Popa', 'Dumitrescu', 'Dumitru', 'Marinescu', 'Anghelescu', 'Stan', 'Rudeanu',
    'Udrea', 'Ciobanu', 'Moise', 'Stoica', 'Moldoveanu', 'Iordache', 'Militaru',
    'Constantinescu', 'Nita', 'Ionita', 'Tudor', 'Dobre', 'Nistor', 'Florea',
    'Stanciu', 'Gheorghiu');

    v_prenume t_denumiri := t_denumiri('Gheorghe', 'Marian', 'Andrei', 'Teodor',
    'Alexandru', 'Marius', 'Costantin', 'Tudor', 'Laurentiu', 'Nicolae', 'Paul',
    'Radu', 'Sorin', 'Toma', 'Traian', 'Vicentiu', 'Ioana', 'Florina', 'Ecaterina',
    'Daniela', 'Carmen', 'Maria', 'Georgeta', 'Sorina', 'Cristina', 'Ana',
    'Luminita', 'Monica', 'Viorica');

    v_telefon NUMBER;
    v_data_ang VARCHAR2(16);
    v_salariu NUMBER;
    v_job VARCHAR2(25);
    v_adresa NUMBER;
    v_depou NUMBER;
    indice_nume NUMBER;
    indice_prenume NUMBER;

BEGIN
    FOR i IN 100..159 LOOP
        indice_nume := random(1, v_nume.count);
        indice_prenume := random(1, v_prenume.count);
        v_telefon := random(700000000, 799999999);
        v_data_ang := random(1, 28) || '/' || random(1, 12) || '/' ||
random(1990, 2014);
        IF i BETWEEN 100 AND 129 THEN v_salariu := random(450, 600) * 10;
        ELSIF i BETWEEN 130 AND 144 THEN v_salariu := random(400, 500) * 10;
        ELSIF i BETWEEN 145 AND 149 THEN v_salariu := random(700, 900) * 10;
        ELSE v_salariu := random(250, 300) * 10;
        END IF;

        IF i BETWEEN 100 AND 129 THEN v_job := 'sofer';
        ELSIF i BETWEEN 130 AND 144 THEN v_job := 'mecanic';
        ELSIF i BETWEEN 145 AND 149 THEN v_job := 'inginer';
        ELSE v_job := 'muncitor necalificat';
        END IF;
        v_adresa := random(1, 60);
        v_depou := (i MOD 7 + 1) * 100 + 1;
        INSERT INTO ANGAJAT VALUES(i, v_nume(indice_nume),
v_prenume(indice_prenume), '+40' || v_telefon, TO_DATE(v_data_ang, 'DD/MM/YYYY'),
v_salariu, v_job, v_adresa, v_depou);
    END LOOP;
END;
/

```

```

IF i BETWEEN 100 AND 129 THEN v_job := 'sofer';
ELSIF i BETWEEN 130 AND 144 THEN v_job := 'mecanic';
ELSIF i BETWEEN 145 AND 149 THEN v_job := 'inginer';
ELSE v_job := 'muncitor necalificat';
END IF;
v_adresa := random(1, 60);
v_depou := (i MOD 7 + 1) * 100 + 1;
INSERT INTO ANGAJAT VALUES(i, v_nume(indice_nume), v_prenume(indice_prenume), '+40' || v_telefon,
TO_DATE(v_data_ang, 'DD/MM/YYYY'), v_salariu, v_job, v_adresa, v_depou);
END LOOP;
END;

```

COD_ANGAJAT	NUME	PRENUME	NUMAR_TELEFON	DATA_ANGAJARE	SALARIU	TIP_JOB	COD_ADRESA	COD_DEPOU
25	124 Rudeanu	Radu	+40707230494	26-DEC-14	5170	sofer	1	601
26	125 Ionescu	Andrei	+40719387961	15-OCT-92	5960	sofer	23	701
27	126 Anghelescu	Luminita	+40712702793	22-FEB-02	4870	sofer	35	101
28	127 Moldoveanu	Maria	+40775593319	20-JAN-94	5750	sofer	10	201
29	128 Ionescu	Alexandru	+40707318059	17-MAR-99	5360	sofer	21	301
30	129 Ciobanu	Alexandru	+40763884505	05-MAR-99	4970	sofer	47	401
31	130 Ionescu	Monica	+40757316407	20-MAY-95	4150	mecanic	57	501
32	131 Grigore	Cristina	+40759441008	28-FEB-96	4160	mecanic	4	601
33	132 Dumitru	Tudor	+40713255206	28-JUN-03	4720	mecanic	46	701
34	123 Branescu	Constantin	+40709022600	15-MAR-04	4350	mecanic	24	101

```

DECLARE
    v_exp          NUMBER;
    v_categ        VARCHAR2(5);
    v_permise      t_denumiri;
    permis_random NUMBER;
    v_data_exp     DATE;
BEGIN
    FOR c IN (SELECT cod_angajat, cod_depou FROM angajat WHERE tip_job = 'sofer')
LOOP
    v_exp := random(5, 40);
    v_permise := t_denumiri();
    IF c.cod_depou IN (101, 301, 401, 501) THEN
        v_permise.extend;
        v_permise(v_permise.last) := 'BUS';
    END IF;
    IF c.cod_depou IN (101, 201, 401, 601) THEN
        v_permise.extend;
        v_permise(v_permise.last) := 'TRL';
    END IF;
    IF c.cod_depou IN (201, 301, 401, 701) THEN
        v_permise.extend;
        v_permise(v_permise.last) := 'TRAM';
    END IF;

    v_data_exp := SYSDATE + 365 * (v_permise.count);
    permis_random := random(1, v_permise.last);
    INSERT INTO SOFER VALUES(c.cod_angajat, v_exp, v_permise(permis_random),
    v_data_exp);
    END LOOP;
END;
/

```

```

DECLARE
    v_meserii  t_denumiri := t_denumiri('Maistru Sudura', 'Maistru Mecanic
Auto', 'Tehnician Retea', 'Tehnician Sudura');
    i          NUMBER      := 1;
    j          NUMBER;
BEGIN
    FOR c IN (SELECT cod_angajat FROM angajat WHERE tip_job = 'mecanic') LOOP
        j := i MOD 4 + 1;
        INSERT INTO SOFER VALUES(c.cod_angajat, v_meserii(j));
        i := i + 1;
    END LOOP;
END;
/

```

```

DECLARE
    v_meserii  t_denumiri := t_denumiri('Maistru Sudura', 'Maistru Mecanic
Auto', 'Tehnician Retea', 'Tehnician Sudura');
    i          NUMBER      := 1;
    j          NUMBER;
BEGIN
    FOR c IN (SELECT cod_angajat FROM angajat WHERE tip_job = 'mecanic') LOOP
        j := i MOD 4 + 1;
        INSERT INTO MECHANIC VALUES(c.cod_angajat, v_meserii(j));
        i := i + 1;
    END LOOP;
END;
/

```

```

DECLARE
    v_meserii  t_denumiri := t_denumiri('Maistru Sudura', 'Maistru Mecanic
Auto', 'Tehnician Retea', 'Tehnician Sudura');
    i          NUMBER      := 1;
    j          NUMBER;
BEGIN
    FOR c IN (SELECT cod_angajat FROM angajat WHERE tip_job = 'mecanic') LOOP
        j := i MOD 4 + 1;
        INSERT INTO MECHANIC VALUES(c.cod_angajat, v_meserii(j));
        i := i + 1;
    END LOOP;
END;
/

```

```

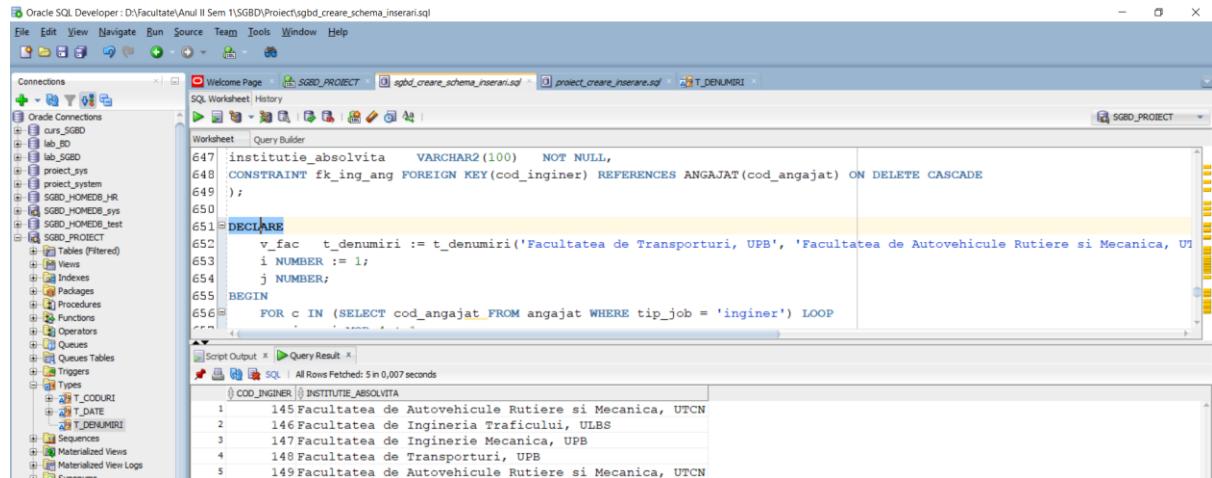
DECLARE
    v_fac  t_denumiri := t_denumiri('Facultatea de Transporturi, UPB',
'Facultatea de Autovehicule Rutiere si Mecanica, UTCN', 'Facultatea de Ingineria
Traficului, ULBS', 'Facultatea de Inginerie Mecanica, UPB');
    i      NUMBER      := 1;
    j      NUMBER;

```

```

BEGIN
    FOR c IN (SELECT cod_angajat FROM angajat WHERE tip_job = 'inginer') LOOP
        j := i MOD 4 + 1;
        INSERT INTO INGINER VALUES(c.cod_angajat, v_fac(j));
        i := i + 1;
    END LOOP;
END;
/

```



```

/*am adaugat o coloana in tabelul TRASEU, care indica tipul de mijloc de
transport utilizat pe fiecare traseu*/
ALTER TABLE TRASEU
ADD vehicule_folosite VARCHAR2(25);
DECLARE
    v_veh    VARCHAR2(25);
BEGIN
    FOR c IN (SELECT numar_traseu FROM traseu) LOOP
        IF c.numar_traseu BETWEEN 1 AND 59 THEN
            v_veh := 'tramvai';
        ELSIF c.numar_traseu BETWEEN 60 AND 99 THEN
            v_veh := 'troleibuz';
        ELSE v_veh := 'autobuz';
        END IF;
        UPDATE traseu t SET t.vehicule_folosite = v_veh WHERE t.numar_traseu =
c.numar_traseu;
    END LOOP;
END;
/
SELECT * FROM TRASEU;

```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the path is: Oracle SQL Developer : D:\Facultate\Anul II Sem 1\SGBD\Project\sbd_create_schema_inserari.sql. The main area displays a query builder window with the following PL/SQL code:

```

701     v_veh := 'troleibuz';
702     ELSE v_veh := 'autobuz';
703   END IF;
704   UPDATE traseu t SET t.vehicule_folosite = v_veh WHERE t.numar_traseu = c.numar_traseu;
705 END LOOP;
706
707 /
708 SELECT * FROM TRASEU;

```

Below the code, a "Query Result" tab is open, showing a table with 15 rows of data. The columns are: NUMAR_TRASEU, LUNGIME, CATEGORIE, FRECENTA, MEDIE_CALATORI, DATA_INFIINTARE, DATA_SUSPENDARE, and VEHICULE_FOLOSITE. The data includes various vehicle types like tramvai, trameibuz, and autobuz.

```

DECLARE
    v_sofer NUMBER;
    v_cod_dep NUMBER;
    v_permis VARCHAR2(5);
    v_tip_veh VARCHAR2(16);
    veh_posibile t_coduri := t_coduri();
    trasee_posibile t_coduri := t_coduri();
    ora1 DATE;
    ora2 DATE;
    ora3 DATE;
    ora4 DATE;
    v_data DATE;
    v_nr_ture NUMBER;
    indice_traseu NUMBER;
    indice_veh NUMBER;
BEGIN
    FOR k IN 1000..1249 LOOP
        v_sofer := random(100, 129);
        veh_posibile := t_coduri();
        trasee_posibile := t_coduri();
        SELECT cod_depou INTO v_cod_dep FROM angajat WHERE cod_angajat = v_sofer;
        SELECT categorie_permis INTO v_permis FROM sofer WHERE cod_sofer =
v_sofer;
        CASE v_permis
            WHEN 'BUS' THEN v_tip_veh := 'autobuz';
            WHEN 'TRL' THEN v_tip_veh := 'troleibuz';
            ELSE v_tip_veh := 'tramvai';
        END CASE;

        /*selectez vehiculele care sunt arondate in acelasi depou in care
lucreaza si soferul si corespund ca tip cu categoria de permis a soferului*/
        SELECT v.cod_vehicul BULK COLLECT INTO veh_posibile
        FROM vehicul v
        JOIN depou d ON d.cod_depou = v.cod_depou
        JOIN model m ON m.cod_model = v.cod_model
        WHERE v.intrebuintare = 'transport'
        AND v.cod_depou = v_cod_dep AND m.tip_mijloc_transport = v_tip_veh;
        indice_veh := random(1, veh_posibile.last);

        /*selectez traseele posibile (sunt trasee pe care se foloseste acelasi
tip de mijloc de transport ca cel pe care il conduce soferul selectat)*/
        SELECT numar_traseu BULK COLLECT INTO trasee_posibile FROM traseu WHERE
        vehicule_folosite = v_tip_veh;
        indice_traseu := random(1, trasee_posibile.last);
        v_data := SYSDATE - random(1, 730);

```

```

        IF trasee_posibile(indice_traseu) <> 901 AND
trasee_posibile(indice_traseu) <> 902 THEN
    ora1 := TO_DATE('04:' || random(30, 59), 'HH24:MI');
    ora2 := TO_DATE('05:0' || random(0, 9), 'HH24:MI');
    ora3 := TO_DATE('22:' || random(10, 59), 'HH24:MI');
    ora4 := TO_DATE('23:' || random(10, 50), 'HH24:MI');

ELSE
    ora1 := TO_DATE('22:' || random(30, 59), 'HH24:MI');
    ora2 := TO_DATE('23:0' || random(0, 9), 'HH24:MI');
    ora3 := TO_DATE('05:' || random(30, 59), 'HH24:MI');
    ora4 := TO_DATE('06:0' || random(0, 9), 'HH24:MI');

END IF;
v_nr_ture := random(7, 25);
INSERT INTO CURSA VALUES(k, v_sofer, veh_posibile(indice_veh),
trasee_posibile(indice_traseu), v_data, ora1, ora2, ora3, ora4, v_nr_ture);
END LOOP;
END;
/
SELECT cod_sofer, cod_vehicul_tr, numar_traseu, data_cursa,
TO_CHAR(ora_preluare, 'hh24:mi') t1, TO_CHAR(ora_incepere_ture, 'hh24:mi') t2,
TO_CHAR(ora_finalizare_ture, 'hh24:mi') t3, TO_CHAR(ora_predare, 'hh24:mi') t4,
numar_ture FROM CURSA;

```

COD_SOFER	COD_VEHICUL_TR	NUMAR_TRASEU	DATA_CURSA	T1	T2	T3	T4	NUMAR_TURE
17	122	1020	381 02-MAR-24 04:38 05:07 22:14 23:15					17
18	121	9088	14 11-MAR-23 04:50 05:09 22:42 23:50					25
19	101	1027	381 23-OCT-23 04:57 05:02 22:40 23:17					19
20	123	1013	381 04-APR-24 04:34 05:00 22:46 23:24					16
21	112	5041	66 01-NOV-23 04:43 05:04 22:24 23:14					23
22	113	5058	73 29-NOV-24 04:43 05:05 22:14 23:11					7
23	114	9079	14 12-JAN-23 04:36 05:03 22:11 23:28					12
24	119	1015	90 12-SEP-24 22:39 23:04 05:30 06:02					15
25	113	5049	85 23-SEP-24 04:31 05:06 22:43 23:37					18
26	106	9093	120 0-JUN-24 04:55 05:05 22:50 23:39					17
27	124	5054	66 21-MAR-24 04:37 05:09 22:27 23:27					15
28	109	1022	90 218-MAY-23 22:45 23:07 05:45 06:02					8
29	119	1007	182 27-APR-24 04:36 05:00 22:28 23:22					12
30	124	5054	66 01-MAR-24 04:58 05:07 22:46 23:26					16

```

CREATE SEQUENCE seq_particip
INCREMENT BY 10 START WITH 10
MAXVALUE 99990;

INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 136, 4, 807);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 133, 4, 807);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 133, 4, 800);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 143, 9, 800);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 139, 9, 800);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 139, 11, 800);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 130, 2, 819);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 131, 2, 819);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 132, 2, 819);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 134, 3, 816);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 142, 10, 814);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 130, 2, 811);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 141, 3, 811);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 134, 6, 801);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 135, 6, 801);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 142, 3, 801);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 138, 7, 804);

```

```
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 138, 3, 806);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 142, 10, 809);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 134, 1, 803);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 137, 1, 803);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 134, 5, 808);
INSERT INTO PARTICIPARE_INTERVENTIE VALUES(seq_particip.nextval, 143, 12, 822);
```

COD_PARTICIPARE	COD_MECHANIC	COD_VEHICULU_INT	NUMAR_INTERVENTIE
1	10	136	4
2	20	133	4
3	30	133	4
4	40	143	9
5	50	139	9
6	60	139	11
7	70	130	2
8	80	131	2
9	90	132	2
10	100	134	3
11	110	142	10
12	120	130	2
13	130	141	3
14	140	144	8

```
INSERT INTO ATELIER VALUES(100, 'U.R.A.C.', TO_DATE('01/06/1998', 'dd/mm/yyyy'), 8, 2);
INSERT INTO ATELIER VALUES(200, 'Atelier Unirii', TO_DATE('01/06/2004', 'dd/mm/yyyy'), 4, 45);
INSERT INTO ATELIER VALUES(300, 'Atelier Berceni', TO_DATE('01/06/1960', 'dd/mm/yyyy'), 12, 3);
INSERT INTO ATELIER VALUES(400, 'Atelier Tital', TO_DATE('01/06/1977', 'dd/mm/yyyy'), 6, 5);
INSERT INTO ATELIER VALUES(500, 'Atelier Vatra Luminoasa', TO_DATE('01/06/1988', 'dd/mm/yyyy'), 4, 11);
```

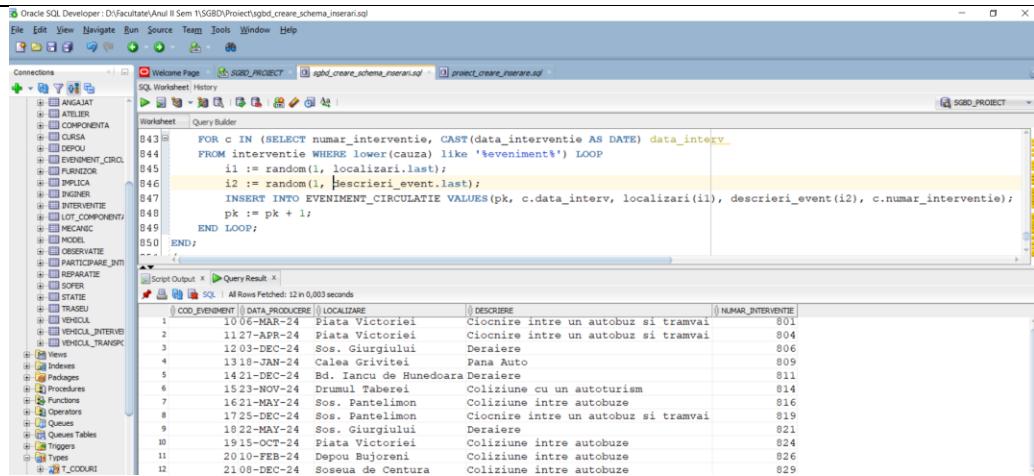
COD_ATELIER	DENUMIRE	AN_INFINTARE	CAPACITATE	COD_ADRESA
100	U.R.A.C.	1998	8	2
200	Atelier Unirii	2004	4	45
300	Atelier Berceni	1960	12	3
400	Atelier Tital	1977	6	5
500	Atelier Vatra Luminoasa	1988	4	11

```
DECLARE
    localizari t_denumiri := t_denumiri('Soseua de Centura', 'Depou Bujoreni',
    'Bd. Nicolae Grigorescu', 'Piata Victoriei', 'Piata Unirii', 'Calea Grivitei',
    'Sos. Giurgiului', 'Drumul Taberei', 'Sos. Pantelimon', 'Bd. Iancu de Hunedoara',
    'Bd. Ion Mihalache');
    descrieri_event t_denumiri := t_denumiri('Deraiere', 'Coliziune cu un
    autoturism', 'Coliziune intre autobuze',
    'Pana Auto', 'Tamponare cu un autoturism', 'Ciocnire intre un autobuz si
    tramvai', 'Incendiu');
```

```

i1 NUMBER;
i2 NUMBER;
pk NUMBER := 10;
BEGIN
    FOR c IN (SELECT numar_interventie, CAST(data_interventie AS DATE)
               data_interv FROM interventie WHERE lower(cauza) like '%eveniment%') LOOP
        i1 := random(1, localizari.last);
        i2 := random(1, descrieri_event.last);
        INSERT INTO EVENIMENT_CIRCULATIE VALUES(pk, c.data_interv,
localizari(i1), descrieri_event(i2), c.numar_interventie);
        pk := pk + 1;
    END LOOP;
END;
/

```



```

INSERT INTO IMPLICAT VALUES(10, 1001, 'Parbriz si Faruri Sparte');
INSERT INTO IMPLICAT VALUES(10, 9084, 'Tampoane indoite, partea din fata a
caroseriei distrusa');
INSERT INTO IMPLICAT VALUES(11, 1008, 'Partea din fata a caroseriei distrusa');
INSERT INTO IMPLICAT VALUES(11, 9072, 'Partea stanga a caroseriei distrusa,
geamuri sparte, articulatia dintre vagoanele 2 si 3 rupta');
INSERT INTO IMPLICAT VALUES(12, 9071, 'Pantograf , Rulmentii boghiurilor
distrusi');

INSERT INTO IMPLICAT VALUES(13, 1030, 'Pneuri sparte');
INSERT INTO IMPLICAT VALUES(14, 9096, 'Pantograf rupt');
INSERT INTO IMPLICAT VALUES(15, 5055, 'Parbriz si faruri sparte');
INSERT INTO IMPLICAT VALUES(16, 1000, 'Aubreiaj stricat');
INSERT INTO IMPLICAT VALUES(16, 1006, 'Caroserie indoita');
INSERT INTO IMPLICAT VALUES(16, 1018, 'Parbriz si Faruri Sparte');
INSERT INTO IMPLICAT VALUES(17, 1001, 'Faruri sparte, caroseria indoita');
INSERT INTO IMPLICAT VALUES(18, 9072, 'Rulmentii boghiurilor distrusi');
INSERT INTO IMPLICAT VALUES(19, 1036, 'Caroseria lateralala distrusa complet');
INSERT INTO IMPLICAT VALUES(19, 1032, 'Parbriz si oglinzi sparte');
INSERT INTO IMPLICAT VALUES(20, 1027, 'Faruri sparte');
INSERT INTO IMPLICAT VALUES(20, 1028, 'Oglinda sparta, caroseria zgariata');
INSERT INTO IMPLICAT VALUES(21, 1016, 'Parbriz crapat');
INSERT INTO IMPLICAT VALUES(21, 1017, 'Oglinda rupta');
INSERT INTO IMPLICAT VALUES(21, 1018, 'Luneta sparta');

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, the connection tree is visible with various schemas like ANGAJAT, ATELIER, and VEHICUL. The central workspace contains a query builder window with the following SQL code:

```

075: INSERT INTO IMPLICIA VALUES(18, 9072, 'Rulmentii boghiurilor distrusi');
076: INSERT INTO IMPLICIA VALUES(19, 1036, 'Caroseria laterala distrusa complet');
077: INSERT INTO IMPLICIA VALUES(19, 1052, 'Parbriz si oglinzi sparte');
078: INSERT INTO IMPLICIA VALUES(20, 1027, 'Faruri sparte');
079: INSERT INTO IMPLICIA VALUES(20, 1028, 'Oglinda sparta, caroseria zgariata');
080: INSERT INTO IMPLICIA VALUES(21, 1016, 'Parbriz crapat');
081: UPDATE TIPURI_TIPURI SET VALOREA=1017 WHERE COD_TIP=1017

```

Below the code, the "Query Result" tab shows the output of the last update statement:

COD_EVENTIMENT	COD_EHOGOLUI_TR	DAUNE
1	10	1001 Parbriz si Faruri Sparte
2	10	9084 Tamponare indoite, partea din fata a caroseriei distrusa
3	11	1008 Partea din fata a caroseriei distrusa
4	11	9072 Partea stanga a caroseriei distrusa, geamuri sparte, articulatia dintre vagoanele 2 si 3 rupta
5	12	9070 Pantograf , Rulmentii boghiurilor distrusi
6	13	1030 Pneuri sparte
7	14	9096 Geamuri sparte
8	15	5055 Parbriz si faruri sparte
9	16	1000 Ambreiaj stricat
10	16	1006 Caroserie indoita
11	16	1018 Parbriz si Faruri Sparte
12	17	1001 Faruri sparte, caroseria indoita
13	18	9072 Rulmentii boghiurilor distrusi
14	19	1036 Caroseria laterala distrusa complet

```

CREATE SEQUENCE seq_reparatii
INCREMENT BY 10 START WITH 10
MAXVALUE 99999
NOCYCLE NOCACHE;
DECLARE
    v_atelier    NUMBER;
    v_inginer    NUMBER;
    v_data        DATE;
BEGIN
    FOR c IN (SELECT cod_vehicul FROM vehicul WHERE stare = 'in reparatie') LOOP
        v_atelier := random(1, 5) * 100;
        v_inginer := random(145, 149);
        v_data := SYSDATE - random(1, 180);
        INSERT INTO REPARATIE VALUES(seq_reparatii.nextval, v_data, null,
c.cod_vehicul, v_inginer, v_atelier);
    END LOOP;
    INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('18-11-2023',
'DD-MM-YYYY'), TO_DATE('01-02-2024', 'DD-MM-YYYY'), 1018, 145, 100);
    INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('18-06-2023',
'DD-MM-YYYY'), TO_DATE('01-08-2023', 'DD-MM-YYYY'), 1018, 145, 100);
    INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('22-02-2023',
'DD-MM-YYYY'), TO_DATE('14-06-2023', 'DD-MM-YYYY'), 9070, 148, 500);
    INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('30-01-2024',
'DD-MM-YYYY'), TO_DATE('01-02-2024', 'DD-MM-YYYY'), 9098, 149, 500);
END;
/

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, the connection tree is visible with various schemas like ANGAJAT, ATELIER, and VEHICUL. The central workspace contains a query builder window with the following SQL code:

```

915: INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('18-06-2023',
'DD-MM-YYYY'), TO_DATE('01-08-2023', 'DD-MM-YYYY'), 1018, 145, 100);
916: INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('22-02-2023',
'DD-MM-YYYY'), TO_DATE('14-06-2023', 'DD-MM-YYYY'), 9070, 148, 500);
917: INSERT INTO REPARATIE VALUES (seq_reparatii.nextval, TO_DATE('30-01-2024',
'DD-MM-YYYY'), TO_DATE('01-02-2024', 'DD-MM-YYYY'), 9098, 149, 500);
918: END;
919:
920:
921:
922:
923:
924:
925:
926:
927:
928:
929:
930:
931:
932:
933:
934:
935:
936:
937:
938:
939:
940:
941:
942:
943:
944:
945:
946:
947:
948:
949:
950:
951:
952:
953:
954:
955:
956:
957:
958:
959:
960:
961:
962:
963:
964:
965:
966:
967:
968:
969:
970:
971:
972:
973:
974:
975:
976:
977:
978:
979:
980:
981:
982:
983:
984:
985:
986:
987:
988:
989:
990:
991:
992:
993:
994:
995:
996:
997:
998:
999:

```

Below the code, the "Query Result" tab shows the output of the last update statement:

COD_REPARATIE	DATA_INCEPERE	DATA_FINALIZARE	COD_VEHICUL_TR	COD_INGINER	COD_ATELIER
9	250 19-SEP-24 (null)		5048	149	200
10	260 26-JUL-24 (null)		5056	146	200
11	270 25-SEP-24 (null)		9063	146	400
12	280 21-NOV-24 (null)		9070	145	300
13	290 21-NOV-24 (null)		9077	149	100
14	300 29-JUL-24 (null)		9084	147	200
15	310 10-NOV-24 (null)		9091	147	300
16	320 22-OCT-24 (null)		9098	149	300
17	330 18-NOV-23 01-FEB-24		1018	145	100
18	340 18-JUN-23 01-AUG-23		1018	145	100
19	350 22-FEB-23 14-JUN-23		9070	148	500
20	360 30-JAN-24 01-FEB-24		9098	149	500

```

INSERT INTO LOT_COMPONENTA VALUES(1, 'Ambreiaj Autobuz', 15, 4500, TO_DATE('18-11-2023', 'DD-MM-YYYY'), 5);

```

```

INSERT INTO LOT_COMPONENTA VALUES(2, 'Parbriz Tramvai', 10, 2100, TO_DATE('10-09-2023', 'DD-MM-YYYY'), 7);
INSERT INTO LOT_COMPONENTA VALUES(3, 'Boghiu', 3, 8500, TO_DATE('18-11-2022', 'DD-MM-YYYY'), 8);
INSERT INTO LOT_COMPONENTA VALUES(4, 'Pneuri Autobuz', 40, 800, TO_DATE('16-10-2023', 'DD-MM-YYYY'), 6);
INSERT INTO LOT_COMPONENTA VALUES(5, 'Caroserie Autobuz', 30, 9900, TO_DATE('18-11-2023', 'DD-MM-YYYY'), 4);
INSERT INTO LOT_COMPONENTA VALUES(6, 'Parbriz Autobuz', 15, 1800, TO_DATE('10-09-2023', 'DD-MM-YYYY'), 7);
INSERT INTO LOT_COMPONENTA VALUES(7, 'Tampoane tramvai', 10, 5000, TO_DATE('10-01-2023', 'DD-MM-YYYY'), 3);
INSERT INTO LOT_COMPONENTA VALUES(8, 'Pantograf tramvai', 5, 9900, TO_DATE('16-09-2022', 'DD-MM-YYYY'), 1);
INSERT INTO LOT_COMPONENTA VALUES(9, 'Faruri Autobuz', 50, 9000, TO_DATE('05-04-2022', 'DD-MM-YYYY'), 2);
INSERT INTO LOT_COMPONENTA VALUES(10, 'Oglinzi Autobuz', 50, 4000, TO_DATE('10-09-2023', 'DD-MM-YYYY'), 7);
INSERT INTO LOT_COMPONENTA VALUES(11, 'Geam Autobuz', 30, 8000, TO_DATE('10-09-2023', 'DD-MM-YYYY'), 8);
INSERT INTO LOT_COMPONENTA VALUES(12, 'Geam Tramvai', 30, 7500, TO_DATE('10-09-2023', 'DD-MM-YYYY'), 8);

```

COD_LOT	NAME_COMPONENTA	CANTITATE	PRET	DATA_LIVRARE	COD_FURNIZOR
1	1 Ambreiaj Autobuz	15	4500	18-NOV-23	5
2	2 Parbriz Tramvai	10	2100	10-SEP-23	7
3	3 Boghiu	3	8500	18-NOV-22	8
4	4 Pneuri Autobuz	40	800	16-OCT-23	6
5	5 Caroserie Autobuz	30	9900	18-NOV-23	4
6	6 Parbriz Autobuz	15	1800	10-SEP-23	7
7	7 Tampoane tramvai	10	5000	10-JAN-23	3
8	8 Pantograf tramvai	5	9900	16-SEP-22	1
9	9 Faruri Autobuz	50	9000	05-APR-22	2
10	10 Oglinzi Autobuz	50	4000	10-SEP-23	7
11	11 Geam Autobuz	30	8000	10-SEP-23	8
12	12 Geam Tramvai	30	7500	10-SEP-23	8

```

INSERT INTO COMPONENTA VALUES(6, 1, 170);
INSERT INTO COMPONENTA VALUES(4, 1, 170);
INSERT INTO COMPONENTA VALUES(4, 2, 170);
INSERT INTO COMPONENTA VALUES(5, 1, 180);
INSERT INTO COMPONENTA VALUES(5, 2, 190);
INSERT INTO COMPONENTA VALUES(6, 2, 200);
INSERT INTO COMPONENTA VALUES(9, 1, 200);
INSERT INTO COMPONENTA VALUES(9, 2, 200);
INSERT INTO COMPONENTA VALUES(11, 1, 210);
INSERT INTO COMPONENTA VALUES(11, 2, 210);
INSERT INTO COMPONENTA VALUES(4, 3, 220);
INSERT INTO COMPONENTA VALUES(4, 4, 220);
INSERT INTO COMPONENTA VALUES(5, 3, 230);
INSERT INTO COMPONENTA VALUES(1, 1, 240);
INSERT INTO COMPONENTA VALUES(1, 2, 250);
INSERT INTO COMPONENTA VALUES(1, 3, 260);
INSERT INTO COMPONENTA VALUES(3, 1, 270);
INSERT INTO COMPONENTA VALUES(3, 2, 280);
INSERT INTO COMPONENTA VALUES(3, 3, 280);
INSERT INTO COMPONENTA VALUES(7, 1, 280);
INSERT INTO COMPONENTA VALUES(7, 2, 280);
INSERT INTO COMPONENTA VALUES(8, 1, 290);
INSERT INTO COMPONENTA VALUES(2, 1, 300);
INSERT INTO COMPONENTA VALUES(7, 3, 300);
INSERT INTO COMPONENTA VALUES(7, 4, 300);
INSERT INTO COMPONENTA VALUES(12, 1, 310);
INSERT INTO COMPONENTA VALUES(12, 2, 310);

```

```
INSERT INTO COMPONENTA VALUES(12, 3, 310);
INSERT INTO COMPONENTA VALUES(2, 2, 320);
INSERT INTO COMPONENTA VALUES(10, 1, 330);
INSERT INTO COMPONENTA VALUES(10, 2, 330);
INSERT INTO COMPONENTA VALUES(6, 3, 340);
INSERT INTO COMPONENTA VALUES(5, 4, 340);
INSERT INTO COMPONENTA VALUES(2, 3, 350);
INSERT INTO COMPONENTA VALUES(8, 2, 360);
```

The screenshot shows the Oracle SQL Developer interface. On the left is the Connections tree, which includes various schema objects like ANGAJAT, ATELIER, CURSA, DEPOLU, EVENIMENT_CIRCOL, FURNIZOR, IMPLICAT, INTERVENIRE, LOT_COMPONENTA, MECANIC, MODEL, OBSERVATIE, PARTICIPARE_INI, PARTNER, SOFER, STATIE, TRASEU, VEHICUL, VEHICUL_INTERV, and VEHICUL_TRANSP. The main area has a 'Welcome Page' tab for 'SGBD_PROJECT'. A 'project_create_inserare.sql' script is open in the 'SQL Worksheet' tab, containing the SQL code shown above. Below it, the 'Query Result' tab displays the results of the last query (line 960) in a grid format:

COD_LOT	NUMAR_COMPONENTA	COD_SEPARATIE
8	9	2 200
9	11	1 210
10	11	2 210
11	4	3 220
12	4	4 220
13	5	3 230
14	1	1 240
15	1	2 250
16	1	3 260
17	3	1 270
18	3	2 280
19	3	3 280
20	7	1 280
21	7	2 280
22	8	1 290
23	2	1 300
24	7	3 300

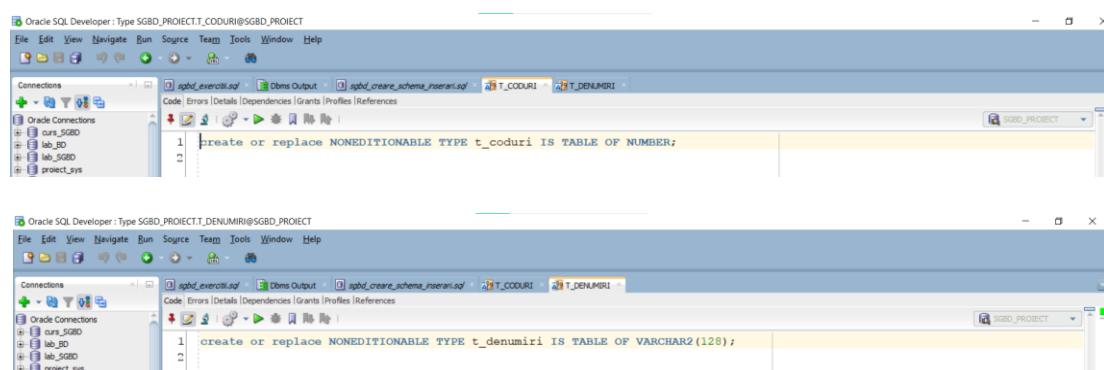
Cerință 6: Colecții

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.

Enunț: Pentru fiecare depou, să se determine și să se afișeze următoarele informații:

- numele și adresa completă.
- gradul de ocupare al depoului, pentru fiecare tip de mijloc de transport care poate fi garat în acel depou (să se afișeze atât sub formă de fracție, cât și procentual), dar și gradul total de ocupare.
- o statistică ce indică starea vehiculelor din depou: câte sunt funcționale, câte defecte și câte se află în prezent în reparație. Să se afișeze numerele de parc (codurile) vehiculelor, iar pentru cele în reparație, data când a început aceasta.
- numărul de angajați per depou, indiferent de job.
- numărul de angajați din fiecare job care lucrează în cadrul depoului.

--observație: tipurile t_coduri și t_denumiri, folosite în următoarele exerciții, au fost definite anterior, fiind folosite și pentru a inseră informații în tabele
 --create or replace NONEDITABLE TYPE t_coduri IS TABLE OF NUMBER;
 --create or replace NONEDITABLE TYPE t_denumiri IS TABLE OF VARCHAR2(128);



```

CREATE OR REPLACE PROCEDURE ex6
IS
    TYPE t_depouri IS TABLE OF depou%rowtype INDEX BY PLS_INTEGER;
    CURSOR cc(p_cod_dep NUMBER) IS
        SELECT v.* , m.tip_mijloc_transport
        FROM vehicul v JOIN model m ON v.cod_model = m.cod_model
        WHERE v.intrebuintare = 'transport'
        AND v.cod_depou = p_cod_dep
        ORDER BY v.cod_vehicul;
    TYPE vehicule_per_categorie IS RECORD
    ( denumire_categ VARCHAR2(32),
      nr NUMBER );
    TYPE t_vehicule IS TABLE OF cc%ROWTYPE;
    TYPE t_vehicule_din_fiecare_tip IS VARRAY(4) OF vehicule_per_categorie;
    TYPE tab_ang IS TABLE OF angajat%rowtype;
    v_vehicule          t_vehicule := t_vehicule();
    v_depouri           t_depouri := t_depouri();
    v_adresa            VARCHAR2(128);
    total_vehicule      NUMBER;
    total_capacitate    NUMBER := 0;
    procent              NUMBER;
    veh_functionale     t_coduri := t_coduri();
    veh_in_reparatie    t_coduri := t_coduri();
    veh_defecte         t_coduri := t_coduri();

```

```

    vect_categorii          t_vehicule_din_fiecare_tip  :=
t_vehicule_din_fiecare_tip();
    categorii                t_denumiri := t_denumiri('autobuz', 'troleibuz',
'tramvai');
    data_intrare_rep         DATE;
    v_angajati               tab_ang := tab_ang();
    total_ang                 NUMBER;
    total_soferi               NUMBER;
    total_inginieri            NUMBER;
    total_mecanici              NUMBER;
    total_muncitori             NUMBER;

FUNCTION numara_categorie (p_vehicule IN t_vehicule, p_categ IN VARCHAR2)
    RETURN NUMBER
IS
    contor NUMBER := 0;
BEGIN
    FOR i IN 1..p_vehicule.LAST LOOP
        IF lower(p_vehicule(i).tip_mijloc_transport) = lower(p_categ) THEN
            contor := contor + 1;
        END IF;
    END LOOP;
    RETURN contor;
END numara_categorie;

PROCEDURE filtreaza_stare (p_vehicule IN t_vehicule, p_stare IN VARCHAR2, rez OUT
NOCOPY t_coduri)
AS
BEGIN
    rez := t_coduri();
    FOR i IN 1..p_vehicule.LAST LOOP
        IF lower(p_vehicule(i).stare) = lower(p_stare) THEN
            rez.extend;
            rez(rez.last) := p_vehicule(i).cod_vehicul;
        END IF;
    END LOOP;
    END filtreaza_stare;

BEGIN
    SELECT * BULK COLLECT INTO v_depouri FROM depou;
    vect_categorii.extend(4);
    FOR i IN 1..v_depouri.last LOOP
        dbms_output.new_line();
        dbms_output.put_line('-----');
        total_vehicule := 0;
        total_ang := 0; total_inginieri := 0; total_soferi := 0; total_mecanici := 0;
        total_muncitori := 0;

        OPEN cc(v_depouri(i).cod_depou);
        FETCH cc BULK COLLECT INTO v_vehicule;
        CLOSE cc;

        SELECT localitate || ', ' || nume_strada || ' nr. ' || numar INTO v_adresa
        FROM adresa WHERE cod_adresa = v_depouri(i).cod_adresa;

        veh_functionale.delete; veh_in_reparatie.delete; veh_defecte.delete;

        FOR k IN 1..categorii.count LOOP
            vect_categorii(k).denumire_categ := categorii(k);
            vect_categorii(k).nr := numara_categorie(v_vehicule, categorii(k));
            total_vehicule := total_vehicule + vect_categorii(k).nr;
        END LOOP;
        total_capacitate := NVL(v_depouri(i).capacitate_autobuze, 0) +
NVL(v_depouri(i).capacitate_troleibuze, 0) + NVL(v_depouri(i).capacitate_tramvaie, 0);
        filtreaza_stare(v_vehicule, 'functional', veh_functionale);
        filtreaza_stare(v_vehicule, 'in reparatie', veh_in_reparatie);
        filtreaza_stare(v_vehicule, 'defect', veh_defecte);
    END LOOP;
END;

```

```

SELECT * BULK COLLECT INTO v_angajati FROM angajat WHERE cod_depou =
v_depouri(i).cod_depou;
    total_ang := v_angajati.count;
    FOR i IN 1..v_angajati.last LOOP
        CASE
            WHEN v_angajati(i).tip_job = 'sofer' THEN
                total_soferi := total_soferi + 1;
            WHEN v_angajati(i).tip_job = 'inginer' THEN
                total_ingineri := total_ingineri + 1;
            WHEN v_angajati(i).tip_job = 'mecanic' THEN
                total_mecanici := total_mecanici + 1;
            ELSE
                total_muncitori := total_muncitori + 1;
        END CASE;
    END LOOP;

    dbms_output.put_line(UPPER(v_depouri(i).denumire) || ' - ' || v_adresa);
    dbms_output.new_line();
    IF v_depouri(i).capacitate_autobuze IS NOT NULL THEN
        procent := vect_categorii(1).nr / v_depouri(i).capacitate_autobuze * 100;
        dbms_output.put_line('Grad de ocupare autobuze: ' || vect_categorii(1).nr
|| '/' || v_depouri(i).capacitate_autobuze
|| ' ' || TRUNC(procent) || '%');
    END IF;

    IF v_depouri(i).capacitate_troleibuze IS NOT NULL THEN
        procent := vect_categorii(2).nr / v_depouri(i).capacitate_troleibuze * 100;
        dbms_output.put_line('Grad de ocupare troleibuze: ' || vect_categorii(2).nr
|| '/' || v_depouri(i).capacitate_troleibuze
|| ' ' || TRUNC(procent) || '%');
    END IF;

    IF v_depouri(i).capacitate_tramvaie IS NOT NULL THEN
        procent := vect_categorii(3).nr / v_depouri(i).capacitate_tramvaie * 100;
        dbms_output.put_line('Grad de ocupare tramvaie: ' || vect_categorii(3).nr
|| '/' || v_depouri(i).capacitate_tramvaie
|| ' ' || TRUNC(procent) || '%');
    END IF;

    procent := total_vehicule / total_capacitate * 100;
    dbms_output.put_line('Grad de ocupare totala: ' || total_vehicule || '/' ||
total_capacitate || ' ' || TRUNC(procent) || '%');
    dbms_output.new_line();

    dbms_output.put_line('Numar vehicule functionale: ' || veh_functionale.count);
    FOR i IN 1..veh_functionale.last LOOP
        dbms_output.put(veh_functionale(i) || ' ');
    END LOOP;
    dbms_output.new_line();

    dbms_output.put_line('Numar vehicule defecte: ' || veh_defecte.count);
    IF veh_defecte.count > 0 THEN
        FOR i IN 1..veh_defecte.last LOOP
            dbms_output.put(veh_defecte(i) || ' ');
        END LOOP;
    END IF;
    dbms_output.new_line();

    dbms_output.put_line('Numar vehicule in reparatie: ' ||
veh_in_reparatie.count);
    IF veh_in_reparatie.count > 0 THEN
        dbms_output.put_line('Vehiculul Data incepere reparatie');
        FOR i IN 1..veh_in_reparatie.last LOOP
            SELECT data_incepere INTO data_intrare_rep
            FROM reparatie r JOIN vehicul_transport vt ON r.cod_vehicul_tr =
vt.cod_vehicul_tr
                WHERE vt.cod_vehicul_tr = veh_in_reparatie(i)
                AND data_finalizare IS NULL;
    END IF;

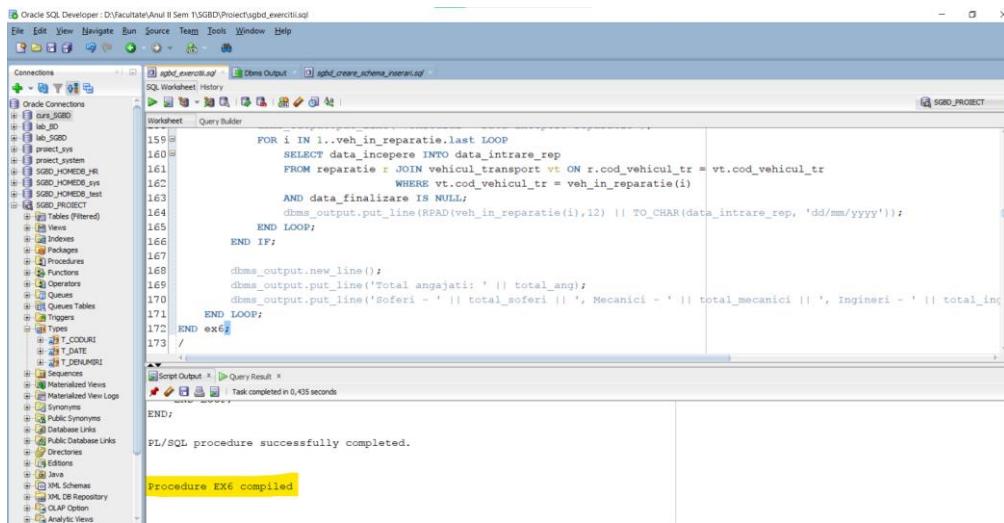
```

```

        dbms_output.put_line(RPAD(veh_in_reparatie(i),12) || 
TO_CHAR(data_intrare_rep, 'dd/mm/yyyy'));
      END LOOP;
    END IF;

    dbms_output.new_line();
    dbms_output.put_line('Total angajati: ' || total_ang);
    dbms_output.put_line('Soferi - ' || total_soferi || ', Mecanici - ' ||
total_mecanici || ', Inginieri - ' || total_inginieri || ', Muncitori necalificati - ' ||
total_muncitori);
  END LOOP;
END ex6;
/

```

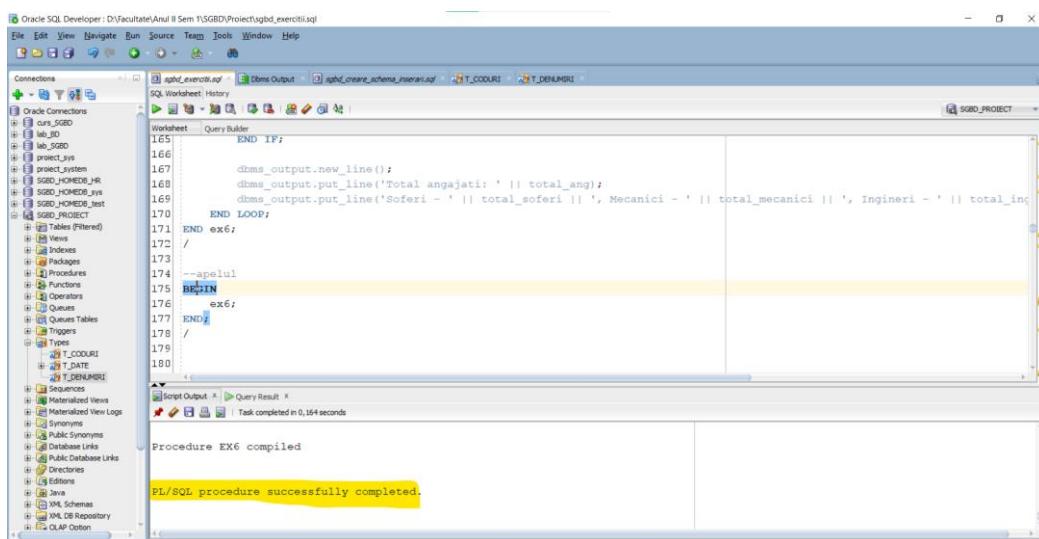


Apelul:

```

BEGIN
  ex6;
END;
/

```



Oracle SQL Developer

```

File Edit View Navigate Run Team Tools Window Help
Connections sgbd_exerciti.sql Dbms Output sgbd_create_schema_inserari.sql T_CODURI
+ Oracle Connections SGBD_PROJECT x
  curs_SGBD
  lab_BO
  lab_SGBD
  project_sys
  project_system
  SGBD_HOMEDB_HR
  SGBD_HOMEDB_sys
  SGBD_HOMEDB_test
  SGBD_PROJECT
Database Schema Service Connection

sgbd_exerciti.sql | Buffer Size:20000 | sgbd_create_schema_inserari.sql | T_CODURI | T_DENUMIRI

SGBD_PROJECT x
-----  

DEPOUL BUJORENTI - Bucuresti, Bd. Timisoara nr. 11  

Grad de ocupare autobuze: 9/20 45%  

Grad de ocupare troleibuze: 6/15 40%  

Grad de ocupare totala: 15/35 42%  

Numar vehicule functionale: 6  

1007 1015 1017 1031 5039 5041  

Numar vehicule defecte: 4  

1008 1016 1032 5055  

Numar vehicule in reparatie: 5  

Vehiculul Data incepere reparatie  

1000 27/10/2024  

1024 15/08/2024  

5040 12/11/2024  

5048 19/09/2024  

5056 26/07/2024  

Total angajati: 8  

Soferi - 4, Mecanici - 2, Inginieri - 1, Muncitori necalificati - 1  

-----  

DEPOUL BUCURESTII NOI - Bucuresti, Bd. Bucurestii Noi nr. 152  

Grad de ocupare troleibuze: 6/30 20%  

Grad de ocupare tramvaie: 12/30 40%  

Grad de ocupare totala: 18/60 30%  

Numar vehicule functionale: 15  

5042 5049 5057 5058 5059 9064 9065 9073 9074 9075 9079 9083 9085 9093 9094 9095  

Numar vehicule defecte: 1  

5050  

Numar vehicule in reparatie: 2  

Vehiculul Data incepere reparatie  

5063 25/04/2024

```

Team Tools Window Help

```

sgbd_exerciti.sql Dbms Output sgbd_create_schema_inserari.sql T_CODURI T_DENUMIRI
+ Oracle Connections SGBD_PROJECT x
  curs_SGBD
  lab_BO
  lab_SGBD
  project_sys
  project_system
  SGBD_HOMEDB_HR
  SGBD_HOMEDB_sys
  SGBD_HOMEDB_test
  SGBD_PROJECT
Database Schema Service Connection

sgbd_exerciti.sql | Buffer Size:20000 | sgbd_create_schema_inserari.sql | T_CODURI | T_DENUMIRI

SGBD_PROJECT x
-----  

Numar vehicule in reparatie: 2  

Vehiculul Data incepere reparatie  

9063 25/09/2024  

9084 29/07/2024  

Total angajati: 8  

Soferi - 4, Mecanici - 2, Inginieri - 1, Muncitori necalificati - 1  

-----  

DEPOUL TITAN - Bucuresti, Bd. Theodor Pallady nr. 7  

Grad de ocupare autobuze: 11/40 27%  

Grad de ocupare tramvaie: 16/20 80%  

Grad de ocupare totala: 27/60 45%  

Numar vehicule functionale: 3  

Vehiculul Data incepere reparatie  

1018 25/12/2024  

9077 21/11/2024  

9098 22/10/2024  

Total angajati: 9  

Soferi - 5, Mecanici - 2, Inginieri - 1, Muncitori necalificati - 1  

-----  

DEPOUL MILITARI - Bucuresti, Str. Preciziei nr. 36  

Grad de ocupare autobuze: 10/20 50%  

Grad de ocupare troleibuze: 6/15 40%  

Grad de ocupare tramvaie: 4/15 26%  

Grad de ocupare totala: 20/50 40%

```

Oracle SQL Developer

```

File Edit View Navigate Run Team Tools Window Help
Connections sgbd_exerciti.sql Dbms Output sgbd_create_schema_inserari.sql T_CODURI
+ Oracle Connections SGBD_PROJECT x
  curs_SGBD
  lab_BO
  lab_SGBD
  project_sys
  project_system
  SGBD_HOMEDB_HR
  SGBD_HOMEDB_sys
  SGBD_HOMEDB_test
  SGBD_PROJECT
Database Schema Service Connection

sgbd_exerciti.sql | Buffer Size:20000 | sgbd_create_schema_inserari.sql | T_CODURI | T_DENUMIRI

SGBD_PROJECT x
-----  

Grad de ocupare totala: 20/50 40%  

Numar vehicule functionale: 15  

1003 1004 1005 1019 1020 1027 1028 1029 5043 5044 5051 5052 5053 9080 9100  

Numar vehicule defecte: 2  

5060 9090  

Numar vehicule in reparatie: 3  

Vehiculul Data incepere reparatie  

1012 24/11/2024  

1036 05/09/2024  

9070 21/11/2024  

Total angajati: 9  

Soferi - 5, Mecanici - 2, Inginieri - 0, Muncitori necalificati - 2  

-----  

DEPOUL FLOREASCA - Bucuresti, Str. Gioachino Rossini nr. 29  

Grad de ocupare autobuze: 9/10 90%  

Grad de ocupare totala: 9/10 90%  

Numar vehicule functionale: 7  

1013 1014 1021 1022 1023 1037 1038  

Numar vehicule defecte: 0  

Numar vehicule in reparatie: 2  

Vehiculul Data incepere reparatie  

1006 08/10/2024  

1030 04/11/2024  

Total angajati: 9  

Soferi - 4, Mecanici - 3, Inginieri - 0, Muncitori necalificati - 2  

-----  

DEPOUL VATRA LUMINOASA - Bucuresti, Str. Tony Bulandra nr. 95

```

Team Tools Window Help

```

sgbd_exerciti.sql Dbms Output sgbd_create_schema_inserari.sql T_CODURI T_DENUMIRI
+ Oracle Connections SGBD_PROJECT x
  curs_SGBD
  lab_BO
  lab_SGBD
  project_sys
  project_system
  SGBD_HOMEDB_HR
  SGBD_HOMEDB_sys
  SGBD_HOMEDB_test
  SGBD_PROJECT
Database Schema Service Connection

sgbd_exerciti.sql | Buffer Size:20000 | sgbd_create_schema_inserari.sql | T_CODURI | T_DENUMIRI

SGBD_PROJECT x
-----  

Total angajati: 9  

Soferi - 4, Mecanici - 3, Inginieri - 0, Muncitori necalificati - 2  

-----  

DEPOUL VATRA LUMINOASA - Bucuresti, Str. Tony Bulandra nr. 95  

Grad de ocupare troleibuze: 6/15 40%  

Grad de ocupare totala: 6/15 40%  

Numar vehicule functionale: 5  

5046 5047 5054 5061 5062  

Numar vehicule defecte: 1  

5045  

Numar vehicule in reparatie: 0  

Total angajati: 9  

Soferi - 4, Mecanici - 2, Inginieri - 1, Muncitori necalificati - 2  

-----  

DEPOUL COLENTINA - Bucuresti, Sos. Colentina nr. 12  

Grad de ocupare tramvaie: 8/30 26%  

Grad de ocupare totala: 8/30 26%  

Numar vehicule functionale: 5  

9071 9081 9082 9092 9101  

Numar vehicule defecte: 2  

9072 9102  

Numar vehicule in reparatie: 1  

Vehiculul Data incepere reparatie  

9091 10/11/2024  

Total angajati: 8  

Soferi - 4, Mecanici - 2, Inginieri - 1, Muncitori necalificati - 1

```

Cerință 7: Cursoare

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.

Enunț: Pentru o lună din an (dată ca parametru), să se determine toți șoferii care au efectuat **minim 2 curse** în acea lună **cu același vehicul** (indiferent de an) și să li se crească salariul cu 5%. Dacă nu există niciun șofer care să respecte condiția, să se afișeze un mesaj sugestiv. Să se determine traseele pe care a circulat șoferul cu acel vehicul, data cursei, intervalul orar în care s-a aflat în cursă și durata medie a unei ture. Pentru acel vehicul, să se menționeze dacă a fost sau nu implicat într-un eveniment de circulație (în caz afirmativ, să se afișeze informații referitoare la fiecare eveniment în care a fost implicat: data producerii, daunele suferite). În plus, să se determine care au fost cele mai circulate trasee (per total) în acea lună.

```

CREATE OR REPLACE PROCEDURE ex7 (p_luna VARCHAR2)
AS
    TYPE refcursor IS REF CURSOR;
    --in cursorul nested, salvez cele mai circulate trasee din acea luna
    CURSOR c_aparitii_lunare IS
        SELECT cod_sofer, cod_vehicul_tr, COUNT(cod_sofer) AS aparitii_pe_masina,
               CURSOR(SELECT numar_traseu
                        FROM cursa c2
                        WHERE TO_CHAR(c2.data_cursa, 'mm') = p_luna
                        GROUP BY numar_traseu HAVING COUNT(numar_traseu) =
                           (SELECT MAX(COUNT(numar_traseu))
                            FROM cursa c3
                            WHERE TO_CHAR(c3.data_cursa, 'mm') = p_luna
                            GROUP BY numar_traseu)
                     )
        FROM cursa
        WHERE TO_CHAR(data_cursa, 'mm') = p_luna
        GROUP BY cod_sofer, cod_vehicul_tr
        ORDER BY 3 DESC, 1;
    --pentru fiecare vehicul, caut evenimentele de circulatie in care a fost
    implicat
    CURSOR vehicul_implicat_accidente(p_cod_vehicul vehicul.cod_vehicul%type) IS
        SELECT ec.data_producere, ec.descriere, i.daune
        FROM implica i, eveniment_circulatie ec
        WHERE i.cod_eveniment = ec.cod_eveniment
        AND i.cod_vehicul_tr = p_cod_vehicul;

    v_cod_sofer      angajat.cod_angajat%type;
    v_cod_veh       vehicul.cod_vehicul%type;
    v_aparitii      NUMBER;
    v_cursor        refcursor;
    v_nume          angajat.nume%type;
    v_prenume       angajat.prenume%type;
    v_salariu_vechi angajat.salariu%type;
    v_salariu_nou  angajat.salariu%type;
    v_traseu        traseu.numar_traseu%type;
    nr_accidente    NUMBER;

BEGIN
    OPEN c_aparitii_lunare;
    LOOP
        FETCH c_aparitii_lunare INTO v_cod_sofer, v_cod_veh, v_aparitii,
        v_cursor;
        --ma opresc cand ajung la soferii care au mai putin de 2 aparitii pe un
        vehicul
        EXIT WHEN v_aparitii < 2 OR c_aparitii_lunare%notfound;
    END LOOP;
    CLOSE c_aparitii_lunare;
END;

```

```

SELECT nume, prenume, salariu INTO v_nume, v_prenume, v_salariu_vechi
    FROM angajat WHERE cod_angajat = v_cod_sofer;

UPDATE angajat
    SET salariu = salariu * 1.05
    WHERE cod_angajat = v_cod_sofer
    RETURNING salariu INTO v_salariu_nou;

dbms_output.put_line('Soferul ' || v_nume || ' ' || v_prenume || ' a
aparut cu vehiculul ' || v_cod_veh || ' pe urmatoarele trasee: ');

FOR cc IN ( SELECT cod_sofer, cod_vehicul_tr, numar_traseu, data_cursa,
ora_incepere_ture, ora_finalizare_ture, TRUNC((ABS(ora_finalizare_ture -
ora_incepere_ture) * 24 * 60) / numar_ture) timp_tura
    FROM cursa
    WHERE cod_sofer = v_cod_sofer AND cod_vehicul_tr = v_cod_veh
AND TO_CHAR(data_cursa, 'mm') = p_luna) LOOP
    dbms_output.put_line(' ->' || cc.numar_traseu || ', in data de ' ||
TO_CHAR(cc.data_cursa, 'dd/mm/yyyy') || ', in intervalul: ' ||
TO_CHAR(cc.ora_incepere_ture, 'hh24:mi') || '-' ||
TO_CHAR(cc.ora_finalizare_ture, 'hh24:mi') || '. Durata medie a unei ture: ' ||
cc.timp_tura || ' minute');
END LOOP;

nr_accidente := 0;
dbms_output.new_line;
dbms_output.put_line('Vehiculul ' || v_cod_veh || ' a fost implicat in
evenimentele de circulatie: ');
FOR event IN vehicul_implicat_accidente(v_cod_veh) LOOP
    nr_accidente := nr_accidente + 1;
    dbms_output.put_line(' -> pe data de ' ||
TO_CHAR(event.data_producere, 'dd/mm/yyyy') || ', ' || event.descriere);
    dbms_output.put_line('         daune suferite: ' || event.daune);
END LOOP;
IF nr_accidente = 0 THEN
    dbms_output.put_line('Nu a fost implicat in niciun eveniment de
circulatie! ');
END IF;

dbms_output.put_line('Salariu vechi: ' || v_salariu_vechi || ' -> Salariu
nou: ' || v_salariu_nou);
dbms_output.put_line('-----');
dbms_output.new_line;
END LOOP;
IF c_aparitii_lunare%rowcount = 1 THEN
    dbms_output.put_line('In luna ' || TO_CHAR(TO_DATE(p_luna, 'mm'),
'MONTH') || ', nu exista niciun sofer care sa fi mers de cel putin 2 ori cu
acelasi vehicul');
END IF;

dbms_output.put('Cele mai circulate trasee in luna ' ||
TO_CHAR(TO_DATE(p_luna, 'mm'), 'MONTH') || ': ');
LOOP
    FETCH v_cursor INTO v_traseu;
    EXIT WHEN v_cursor%notfound;
    dbms_output.put(v_traseu || ' ');
END LOOP;
dbms_output.new_line;
CLOSE v_cursor;
CLOSE c_aparitii_lunare;
END ex7;
/

```

```

spbd_exercitii.sql  Dmbs Output  sgbd_create_schema_inserari.sql  Welcome Page  SGBD_PROJECT
File Edit View Navigate Run Source Team Tools Window Help
Connections Oracle Connections
+ curs_SGDB
+ lab_BD
+ lab_SGDB
+ project_sys
+ project_system
+ SGBD_HOMEDEV_HR
+ SGBD_HOMEDEV_sys
+ SGBD_HOMEDEV_test
+ SGBD_PROJECT
Tables (Filtered)
Views
Indexes
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editors
Java
XML Schemas
XML DB Repository
OLAP Option
Analytic Views
Scheduler
Property Graph
RDP Semantic Graph
Recycle Bin
Script Output: Task completed in 0,969 seconds
Procedure EX7 compiled
Messages - Log
Messages Logging Page Statements

```

Exemplu de apel:

```

BEGIN
    ex7('01');
END;
/
--rollback;

```

```

spbd_exercitii.sql  Dmbs Output  sgbd_create_schema_inserari.sql  Welcome Page  SGBD_PROJECT
File Edit View Navigate Run Team Tools Window Help
Connections Oracle Connections
+ curs_SGDB
+ lab_BD
+ lab_SGDB
+ project_sys
+ project_system
+ SGBD_HOMEDEV_HR
+ SGBD_HOMEDEV_sys
+ SGBD_HOMEDEV_test
+ SGBD_PROJECT
Tables (Filtered)
Views
Indexes
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editors
Java
XML Schemas
XML DB Repository
OLAP Option
Soferul Ionescu Sorina a aparut cu vehiculul 1001 pe urmatoarele trasee:
->104, in data de 11/01/2023, in intervalul: 05:04-22:43. Durata medie a unei ture: 62 minute
->605, in data de 19/01/2023, in intervalul: 05:05-22:23. Durata medie a unei ture: 45 minute

Vehiculul 1001 a fost implicat in evenimentele de circulatie:
-> pe data de 06/03/2024, Ciocnire intre un autobuz si tramvai
    daune suferite: Parbriz si Faruri Sparte
-> pe data de 25/12/2024, Ciocnire intre un autobuz si tramvai
    daune suferite: Faruri sparte, caroseria indoita
Salariu vechi: 5576 -> Salariu nou: 5855
-----

Soferul Moldoveanu Traian a aparut cu vehiculul 5045 pe urmatoarele trasee:
->85, in data de 03/01/2024, in intervalul: 05:04-22:29. Durata medie a unei ture: 58 minute
->66, in data de 09/01/2024, in intervalul: 05:02-22:36. Durata medie a unei ture: 150 minute

Vehiculul 5045 a fost implicat in evenimentele de circulatie:
Nu a fost implicat in niciun eveniment de circulatie!
Salariu vechi: 6048 -> Salariu nou: 6350
-----

Cele mai circulate trasee in luna JANUARY : 85 73

```

```

BEGIN
    ex7('07');
END;
/
--rollback;

```

```
In luna JULY , nu exista niciun sofer care sa fi mers de cel putin 2 ori cu acelasi vehicul
Cele mai circulate trasee in luna JULY : 85 73 21
```

```
BEGIN
    ex7('09');
END;
/
--rollback;
```

```
Soferul Florea Maria a aparut cu vehiculul 1038 pe urmatoarele trasee:
->605, in data de 08/09/2023, in intervalul: 05:02-22:23. Durata medie a unei ture: 41 minute
->901, in data de 09/09/2023, in intervalul: 23:04-05:37. Durata medie a unei ture: 61 minute

Vehiculul 1038 a fost implicat in evenimentele de circulatie:
Nu a fost implicat in niciun eveniment de circulatie!
Salariu vechi: 5170 -> Salariu nou: 5429
-----
Soferul Popescu Luminita a aparut cu vehiculul 1000 pe urmatoarele trasee:
->901, in data de 08/09/2023, in intervalul: 23:09-05:35. Durata medie a unei ture: 62 minute
->901, in data de 27/09/2024, in intervalul: 23:02-05:41. Durata medie a unei ture: 49 minute

Vehiculul 1000 a fost implicat in evenimentele de circulatie:
-> pe data de 21/05/2024, Coliziune intre autobuze
daune suferite: Ambreiaj stricat
Salariu vechi: 4620 -> Salariu nou: 4851
-----
Cele mai circulate trasee in luna SEPTEMBER: 901
```

Cerință 8: Subprogram stocat de tip funcție, tratarea exceptiilor

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 create. Tratați toate exceptiile care pot apărea, incluzând exceptiile predefinite NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați **toate** cazurile tratate.

Enunț: Pentru un nume de stație introdus de la tastatură, să se găsească **câte din traseele active tranzitează acea stație** și să se afișeze indicativele acestor trasee, împreună cu numele stațiilor de la capetele liniei. Pentru fiecare dintre traseele determinate, să se afle **câte modele** diferite de vehicule au circulat pe acel traseu în **ultimele 3 luni**. Să se trateze cazurile: există mai multe stații cu numele introdus, nu există nicio stație cu numele introdus, o stație nu apare pe niciun traseu.

(Valoarea returnată de funcție reprezintă numărul de trasee al căror itinerariu conține stația dată ca parametru, în timp ce pentru celelalte informații cerute, referitoare la trasee, se va folosi un parametru de tip tablou de înregistrări. Afisarea informațiilor se va face în blocul în care se apelează funcția.)

```

CREATE OR REPLACE TYPE rec_traseu IS OBJECT
(
    indicativ_traseu      NUMBER,
    statie_capat_1         VARCHAR2(64),
    statie_capat_2         VARCHAR2(64),
    nr_modele              NUMBER
);
/
CREATE OR REPLACE TYPE tab_trasee IS TABLE OF rec_traseu;
/
CREATE OR REPLACE FUNCTION ex8 (v_numere_statie statie.nume%type, v_trasee OUT
tab_trasee)
RETURN NUMBER
IS
    nr_trasee      NUMBER;
    v_statie        statie%rowtype;
    nu_apare_pe_trasee EXCEPTION;
    v_coduri_statii t_coduri;
    v_numere_statii t_denumiri;
    contine_statia BOOLEAN;
    v_indicativ     traseu.numar_traseu%type;
    v_capat_1       statie.nume%type;
    v_capat_2       statie.nume%type;
    v_nr_modele    NUMBER;
    CURSOR itinerariu_traseu(param_traseu    traseu.numar_traseu%type) IS
        SELECT pt.cod_statie, s.numere
        FROM parcurs_traseu pt JOIN statie s ON pt.cod_statie = s.cod_statie
        WHERE pt.numar_traseu = param_traseu
        ORDER BY pt.numar_ordine;

BEGIN
    v_trasee := tab_trasee();
    SELECT * INTO v_statie FROM statie WHERE lower(nume) LIKE '%' || lower(v_numere_statie) || '%';

    --numar pe care trasee (active) apare statia data ca parametru
    SELECT COUNT(*) INTO nr_trasee

```

```

    FROM parcurs_traseu pt
    JOIN traseu t ON pt.numar_traseu = t.numar_traseu
    WHERE t.data_suspendare IS NULL
    AND pt.cod_statie = v_statie.cod_statie;

    IF nr_trasee = 0 THEN
        RAISE nu_apare_pe_trasee;
    END IF;

    --parcure traseele si le selectez doar pe cele care au in itinerariul lor
    statia ceruta
    FOR c IN (SELECT numar_traseu FROM traseu WHERE data_suspendare IS NULL) LOOP
        contine_statia := FALSE;
        OPEN itinerariu_traseu(c.numar_traseu);
        FETCH itinerariu_traseu BULK COLLECT INTO v_coduri_statii, v_nume_statii;
        CLOSE itinerariu_traseu;
        FOR i IN 1..v_coduri_statii.last LOOP
            IF v_coduri_statii(i) = v_statie.cod_statie THEN
                contine_statia := TRUE;
            END IF;
        END LOOP;
        IF contine_statia = TRUE THEN
            v_trasee.extend;
            v_indicativ := c.numar_traseu;
            v_capat_1 := v_nume_statii(v_nume_statii.first);
            v_capat_2 := v_nume_statii(v_nume_statii.last);

            --calculez cate modele diferite au aparut pe acest traseu in ultimele
3 luni
            SELECT COUNT (DISTINCT m.denumire) INTO v_nr_modele
            FROM cursa cr
                JOIN vehicul_transport vt ON vt.cod_vehicul_tr = cr.cod_vehicul_tr
                JOIN vehicul v ON v.cod_vehicul = vt.cod_vehicul_tr
                JOIN model m ON v.cod_model = m.cod_model
            WHERE cr.numar_traseu = c.numar_traseu AND MONTHS_BETWEEN(SYSDATE,
            cr.data_cursa) <= 3;

            --adaug o noua inregistrare in tablou
            v_trasee(v_trasee.last) := rec_traseu(v_indicativ, v_capat_1,
            v_capat_2, v_nr_modele);
            END IF;
        END LOOP;
        RETURN nr_trasee;
    EXCEPTION
        WHEN TOO_MANY_ROWS THEN
            RAISE_APPLICATION_ERROR(-20001, 'Exista mai multe statii cu acest nume');
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20002, 'Nu exista nicio statie cu acest nume');
        WHEN nu_apare_pe_trasee THEN
            RAISE_APPLICATION_ERROR(-20003, 'Statia data nu apare pe niciun traseu');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Codul erorii: ' || SQLCODE);
            DBMS_OUTPUT.PUT_LINE('Mesajul erorii: ' || SQLERRM);
    END;
/

```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database connections and objects. The right pane shows the code for the function EX8:

```

396      --adaug o noua inregistrare in tablou
397      v_trasee(v_trasee.last) := rec_traseu(v_indicativ, v_capat_1, v_capat_2, v_nr_modele);
398  END IF;
399  END LOOP;
400  RETURN nr_trasee;
401 EXCEPTION
402  WHEN TOO_MANY_ROWS THEN
403    RAISE_APPLICATION_ERROR(-20001, 'Există mai multe stații cu acest nume');
404  WHEN NO_DATA_FOUND THEN
405    RAISE_APPLICATION_ERROR(-20002, 'Nu există nicio stație cu acest nume');
406  WHEN nu_apare_pe_trasee THEN
407    RAISE_APPLICATION_ERROR(-20003, 'Stația data nu apare pe niciun traseu');
408  WHEN OTHERS THEN
409    DBMS_OUTPUT.PUT_LINE('Codul erorii: ' ||SQLCODE||);
410    DBMS_OUTPUT.PUT_LINE('Mesajul erorii: ' ||SQLERRM||);
411  END;
412 /

```

Below the code, it says "Function EX8 compiled".

```

--apelul functiei si afisarile
DECLARE
  v_statie      statie.numere%type  := '&p_statie';
  nr_trasee     NUMBER;
  trasee        tab_trasee         := tab_trasee();
BEGIN
  nr_trasee   := ex8(v_statie, trasee);
  dbms_output.put_line('Stația ' || initcap(v_statie) || ' apare pe ' ||
nr_trasee || ' trasee:');
  FOR i IN 1..trasee.last LOOP
    dbms_output.put_line(trasee(i).indicativ_traseu || ':' || |
trasee(i).stati_capat_1 || ' -> ' || trasee(i).stati_capat_2);
    dbms_output.put_line(' Pe acest traseu au circulat ' ||
trasee(i).nr_modele || ' modele de vehicule in ultimele 3 luni.');
  END LOOP;
END;
/

```

Exemplu 1: cazul în care nu se aruncă **nicio excepție** (parametru de intrare: ‘Doamna Ghica’)

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database connections and objects. The right pane shows the code for the function EX8, with a substitution variable dialog open:

```

414
415  --apelul
416  DECLARE
417    v_statie      statie.numere%type  := '&p_statie';
418    nr_trasee     NUMBER;
419    trasee        tab_trasee         := tab_trasee();
420  BEGIN
421    nr_trasee   := ex8(v_statie);
422    dbms_output.put_line('Stația Doamna Ghica apare pe ' || nr_trasee || ' trasee:');
423  FOR i IN 1..trasee.last LOOP
424    dbms_output.put_line(trasee(i).indicativ_traseu || ':' || |
trasee(i).stati_capat_1 || ' -> ' || trasee(i).stati_capat_2);
425    dbms_output.put_line(' Pe acest traseu au circulat ' ||
trasee(i).nr_modele || ' modele de vehicule in ultimele 3 luni.');
426  END LOOP;
427
428 /

```

A modal dialog titled "Enter Substitution Variable" is open, asking for the value of p_statie. The input field contains "Doamna Ghica".

```

Connections
  Oracle Connections
    curs_SGBD
    lab_80
    lab_SGBD
    project
    project_system
    SGBD_HOMEDB_HR
    SGBD_HOMEDB_sys
    SGBD_HOMEDB_test
    SGBD_PROJECT
      Tables (Filtered)
      Views
      Indexes
      Packages
      Procedures

sgbd_exercitii.sql  Dbsm Output  sgbd_create_schema_inserari.sql  Welcome Page  SGBD_PROJECT  STATISTICI_CURSE_LUNARE

SGBD_PROJECT
Statia Doamna Ghica apare pe 4 trasee:
1: Universitate -> Granitul
  Pe acest traseu au circulat 2 modele de vehicule in ultimele 3 luni.
73: Stadionul National -> Mihai Eminescu
  Pe acest traseu au circulat 3 modele de vehicule in ultimele 3 luni.
85: Vasile Lascăr -> Lizeanu
  Pe acest traseu au circulat 3 modele de vehicule in ultimele 3 luni.
902: Pacii -> Doamna Ghica
  Pe acest traseu au circulat 0 modele de vehicule in ultimele 3 luni.

```

Exemplu 2: excepția **too_many_rows** (parametrul de intrare: ‘Piata Unirii’ – există stațiile Piata Unirii 1 și Piata Unirii 2)

```

sgbd_exercitii.sql  Dbsm Output  sgbd_create_schema_inserari.sql  Welcome Page  SGBD_PROJECT  STATISTICI_CURSE_LUNARE

SQL Worksheet History

Worksheet Query Builder

414
415 --apelul
416 DECLARE
417   v_statie      statie.numere%type  := '&p_statie';
418   nr_trasee     NUMBER;
419   trasee        tab_trasee();
420 BEGIN
421   nr_trasee  := ex8(v_statie);
422   dbms_output.put_line('Statia ' || initcap(v_statie) || ' apare pe ' || nr_trasee || ' trasee:');
423   FOR i IN 1..trasee.last
424     dbms_output.put_line('  ' || trasee(i).statie_capat_1 || ' -> ' || trasee(i).statie_capat_2);
425   dbms_output.put_line('');
426   END LOOP;
427 END;
428 /
429
430 -- Doamna Ghica (ok)
431 -- Piata Unirii (too many rows)

Enter Substitution Variable X
Enter value for p_statie:
Piata Unirii
OK Cancel

```

```

sgbd_exercitii.sql  Dbsm Output  sgbd_create_schema_inserari.sql  Welcome Page  SGBD_PROJECT  STATISTICI_CURSE_LUNARE

SQL Worksheet History

Worksheet Query Builder

418   nr_trasee     NUMBER;
419   trasee        tab_trasee();
420 BEGIN
421   nr_trasee  := ex8(v_statie, trasee);
422   dbms_output.put_line('Statia ' || initcap(v_statie) || ' apare pe ' || nr_trasee || ' trasee:');
423   FOR i IN 1..trasee.last LOOP
424     dbms_output.put_line(trasee(i).indicativ_traseu || ':' || trasee(i).statie_capat_1 || ' -> ' || trasee(i).statie_capat_2);
425   dbms_output.put_line('  Pe acest traseu au circulat ' || trasee(i).nr_modele || ' modele de vehicule in ultimele 3 luni');
426   END LOOP;
427 END;
428 /
429
430 -- Doamna Ghica (ok)

Script Output X | Query Result X
Task completed in 19,303 seconds
dbms_output.put_line('  Pe acest traseu au circulat ' || trasee(i).nr_modele || ' modele de vehicule in ultimele 3 luni');
END LOOP;
END;
Error report -
ORA-20001: Există mai multe stații cu acest nume
ORA-06512: at "SGBD_PROJECT.EX8", line 67
ORA-06512: at line 6

4

```

Exemplu 3: exceptia **no_data_found** (parametrul de intrare: ‘Aeroport’)

The screenshot displays two Oracle SQL Developer windows. The top window shows a PL/SQL block being run, specifically line 434 which contains a substitution variable '&p_statie'. A modal dialog box titled 'Enter Substitution Variable' is open, prompting for a value. The user has entered 'Aeroport' in the input field. The bottom window shows the execution results, including the output of the PL/SQL block and an error report at the end.

```
424     dbms_output.put_line(trasee(i).indicativ_traseu || ':' || trasee(i).statie_capat_1 || ' -> ' || trasee(i).
425     dbms_output.put_line('    Pe acest traseu au circulat ' || trasee(i).nr_modele || ' modele de vehicule in
426   END LOOP;
427
428 /
429
430 -- Doamna Ghica (ok)
431 -- Piata Unirii (too many)
432 -- Precizie (fara trasee)
433 -- Aeroport (no data found)
434

411 END;
412 /
413
414
415 --apelul
416 DECLARE
417   v_statie      statie.nume%type  := '&p_statie';
418   nr_trasee    NUMBER;
419   trasee       tab_trasee      := tab_trasee();
420 BEGIN
421   nr_trasee  := ex8(v_statie, trasee);

dbms_output.put_line(trasee(i).indicativ_traseu || ':' || trasee(i).statie_capat_1 || ' -> ' || trasee(i).statie_capat_2);
dbms_output.put_line('    Pe acest traseu au circulat ' || trasee(i).nr_modele || ' modele de vehicule in ultimul
END LOOP;
END;
Error report -
ORA-20002: Nu exista nicio statie cu acest nume
ORA-06512: at "SGBD_PROJECT.EX8", line 69
ORA-06512: at line 6
```

Exemplu 4: excepția `nu_apare_pe_trasee` (excepție custom) (parametrul de intrare: ‘Preciziei’)

Worksheet Query Builder

```
424     dbms_output.put_line(trasee(i).indicativ_traseu || ':' || trasee(i).statie_capat_1 || ' -> ' || trasee(i).
425     dbms_output.put_line('    Pe acest traseu au circulat ' || trasee(i).nr_modele || ' modele de vehicule in
426     END LOOP;
427 END;
428 /
429
430 -- Doamna Ghica (ok)
431 -- Piata Unirii (too many
432 -- Preciziei (fara trasee)
433 -- Aeroport (no data found)
434
```

Enter Substitution Variable

Enter value for p_statie:

Preciziei

OK Cancel

```

411 END;
412 /
413
414
415 --apelul
416 DECLARE
417   v_statie      statie.numere%type := '&p_statie';
418   nr_trasee     NUMBER;
419   trasee        tab_trasee         := tab_trasee();
420 BEGIN
421   nr_trasee    := ex8(v_statie, trasee);
422
423   FOR i IN 1 .. nr_trasee LOOP
424     dbms_output.put_line(trasee(i).indicativ_traseu || ':' || trasee(i).statie_capat_1 || ' -> ' || trasee(i).statie_capat_2);
425     dbms_output.put_line(' Pe acest traseu au circulat ' || trasee(i).nr_modele || ' modele de vehicule in ultimii 6 luni');
426   END LOOP;
427
428 END;
429
430 Error report -
431 ORA-20003: Statia data nu apare pe niciun traseu
432 ORA-06512: at "SGBD_PROJECT.EX8", line 71
433 ORA-06512: at line 6

```

COD_STATIE	NUME	ARTERA	LOCALITATE
1	1004 Bd. Ferdinand	Bucuresti Bd. Carol I	
2	1020 Bd. Laminorului	Bucuresti Str. Turnu Magurele	
3	1035 Favorit	Bucuresti Sos. Pantelimon	
4	1036 Orizont	Bucuresti Bd. Carol I	
5	1041 Timpuri Noi	Bucuresti Str. Barbu Vacarescu	
6	1044 Pasaj Colentina	Bucuresti Str. Turnu Magurele	
7	1057 Valea Cascadelor	Bucuresti Sos. Pantelimon	
8	1058 Preciziei	Bucuresti Bd. Carol I	
9	2001 Gladiolelor	Domenesti Str. Eroilor	
10	2009 Soseaua de Centura	Bragadiru Str. Eroilor	
11	2010 Cartierul Latin	Domenesti Str. Arcului	

Pentru a obtine toate statiile care nu apar pe niciun traseu:

```

select * from statie
where cod_statie not in
(select cod_statie from
parcurs_traseu);

```

Cerință 9: Subprogram stocat de tip procedură, excepții proprii

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un **subprogram stocat independent de tip procedură** care să aibă minim 2 parametri și să utilizeze **într-o singură comandă SQL** 5 dintre tabelele create. Definiți minim 2 excepții proprii, altele decât cele predefinite la nivel de sistem. Apelați subprogramul astfel încât să evidențiați **toate** cazurile definite și tratate.

Enunț:

Se prelungesc lucrările de reparație ale unui vehicul și mai este necesar un anumit număr de bucăți dintr-o componentă. Dacă mai sunt suficiente piese pe stoc, **indiferent de lotul de proveniență**, să se utilizeze la reparație. Dacă există mai multe loturi cu aceeași piesă, mai întâi se vor consuma componentele din lotul mai vechi.

Apoi, să se încerce, **în limita locurilor disponibile**, mutarea vehiculului aflat sub reparație într-un alt atelier, care să se afle **la aceeași adresă cu un depou**. Mutarea se va face în acel atelier care are mai multe locuri libere. (Să se afișeze un mesaj care să indice dacă s-a efectuat sau nu mutarea). Coordonarea reparației va fi preluată de către inginerul care lucrează în depoul aflat la aceeași adresă cu atelierul în care s-a realizat mutarea.

Să se afișeze informații referitoare la reparație, **înainte și după actualizări**: numele atelierului în care se află, inginerul coordonator, producătorul vehiculului și modelul, vechimea mijlocului de transport, numărul de componente schimbate în acea reparație.

(Observații: în fiecare depou, există cel mult un inginer, iar depourile care conțin și atelier au exact un inginer. De asemenea, am presupus, ca o regulă a modelului, că un anumit tip de componentă este compatibil cu orice model, adică, de exemplu, un boghiu se potrivește la orice tramvai.)

Depourile care au și atelier (ambele se află la aceeași adresă):

```
SELECT d.denumire den_depou, at.denumire den_atelier, d.cod_depou, at.cod_atelier
FROM depou d JOIN adresa a ON d.cod_adresa = a.cod_adresa
JOIN atelier at ON a.cod_adresa = at.cod_adresa;
```

DEN_DEPOU	DEN_ATELIER	COD_DEPOU	COD_ATELIER
1 Depoul Titan	Atelier Tital	301	400
2 Depoul Vatra Luminoasa	Atelier Vatra Luminoasa	601	500

```
--inserturi pentru a avea mai multe loturi cu aceeași componentă
INSERT INTO lot_componenta VALUES (13, 'Tampoane tramvai', 15, 8800, SYSDATE, 1);
INSERT INTO lot_componenta VALUES (14, 'Boghiu', 1, 700, SYSDATE, 8);
INSERT INTO componenta VALUES(13, 1, 300);
INSERT INTO componenta VALUES(13, 2, 300);
```

Procedura va primi ca parametri: numele componentei necesare reparației, numărul de bucăți necesare și codul reparației.

```

CREATE OR REPLACE PROCEDURE ex9
(v_num_piesa lot_componenta.nume_componenta%type,
v_nr_buc lot_componenta.cantitate%type,
v_cod_reparatie reparatie.cod_reparatie%type)
IS
    piese_insuficiente EXCEPTION;
    reparatie_finalizata EXCEPTION;
    deja_in_atelier EXCEPTION;
    ateliere_piese EXCEPTION;
    CURSOR calcul_stoc_piesa(param_nume_piesa
lot_componenta.nume_componenta%type) IS
        SELECT l.cod_lot, COUNT(c.cod_lot) "consumate",
               (SELECT 12.cantitate FROM lot_componenta 12 WHERE 12.cod_lot =
l.cod_lot) AS "stoc initial",
               (SELECT 13.data_livrare FROM lot_componenta 13 WHERE 13.cod_lot =
l.cod_lot)
        FROM lot_componenta l LEFT JOIN componenta c ON l.cod_lot =
c.cod_lot
        WHERE LOWER(l.nume_componenta) = LOWER(param_nume_piesa)
        GROUP BY l.cod_lot
        ORDER BY 4;
    CURSOR depou_cu_atelier IS
        SELECT d.cod_depou, at.cod_atelier, at.capacitate,
               at.capacitate - (SELECT COUNT(*) FROM reparatie r2 WHERE
r2.data_finalizare IS NULL AND r2.cod_atelier = at.cod_atelier) locuri_libere
        FROM depou d JOIN adresa a ON d.cod_adresa = a.cod_adresa
        JOIN atelier at ON a.cod_adresa = at.cod_adresa
        ORDER BY 4 DESC;

    TYPE rec_stoc IS RECORD
    (
        cod_lot          lot_componenta.cod_lot%type,
        consumate       lot_componenta.cantitate%type,
        buc_livrare     lot_componenta.cantitate%type,
        data_livrare    lot_componenta.data_livrare%type
    );
    TYPE tab_stoc IS TABLE OF rec_stoc;
    v_stocuri      tab_stoc      := tab_stoc();
    buc_consumate   NUMBER := 0;
    v_total_piese   NUMBER;
    v_consumate_din_lot NUMBER;
    piese_adaugate  NUMBER;
    piese_adaugate_din_lot NUMBER;
    poz            NUMBER;
    nr_ord_piese    NUMBER;
    v_data_fin      DATE;
    v_atelier_vechi reparatie.cod_atelier%type;
    v_atelier_nou   reparatie.cod_atelier%type;
    v_inginer_nou   reparatie.cod_inginer%type;
    v_depou         depou.cod_depou%type;
    v_capacitate    NUMBER;
    v_locuri_libere NUMBER;

    PROCEDURE afiseaza_mesaj(p_cod_reparatie reparatie.cod_reparatie%type)
    IS
    BEGIN
        FOR c IN(
            SELECT TO_CHAR(rep.data_incepere, 'dd.mm.yyyy') data, v.producator,
m.denumire den_model, TRUNC((SYSDATE - v.an_achizitie) / 365) ani_vech, a.nume ||
' ' || a.prenume suprvisor, at.denumire den_atelier, COUNT(c.cod_reparatie)
nr_piese_inloc
            FROM reparatie rep
            JOIN inginer i ON i.cod_inginer = rep.cod_inginer
            JOIN angajat a ON a.cod_angajat = i.cod_inginer
            JOIN atelier at ON at.cod_atelier = rep.cod_atelier
            JOIN vehicul_transport vt ON vt.cod_vehicul_tr = rep.cod_vehicul_tr
            JOIN vehicul v ON v.cod_vehicul = vt.cod_vehicul_tr
        )
        LOOP
            dbms_output.put_line('Mesaj: ' || c.data || ' | ' || c.v_producator || ' | ' || c.m_denumire || ' | ' || c.v_an_achizitie || ' | ' || c.a_nume || ' ' || c.a_prenume || ' | ' || c.at_denumire || ' | ' || c.nr_piese_inloc);
        END LOOP;
    END;

```

```

        JOIN model m ON m.cod_model = v.cod_model
        JOIN componenta c ON c.cod_reparatie = rep.cod_reparatie
        WHERE rep.cod_reparatie = p_cod_reparatie
        GROUP BY TO_CHAR(rep.data_incepere, 'dd.mm.yyyy'), v.producator,
m.denumire, TRUNC((SYSDATE - v.an_achizitie) / 365), a.nume || ' ' || a.prenume,
at.denumire)
    LOOP
        dbms_output.put_line('Vehiculul ' || c.den_model || ' cu o vechime de
' || c.anii_vechi || ' ani, produs de ' ||
c.producator || ' se afla in reparatie din data de ' || c.data || '
la ' || c.den_atelier || '. ');
        dbms_output.put_line('Reparatia este coordonata de catre Ing. ' ||
c.supervizor || '. ');
        dbms_output.put_line('Pana la acest moment, s-au inlocuit ' ||
c.nr_piese_inloc || ' piese.');
        dbms_output.new_line;
    END LOOP;
END afiseaza_mesaj;
BEGIN
/*pentru a putea trata individual exceptiile no_data_found (nu exista
reparatie cu codul dat / piesa cu numele dat), am inclus acest select...into care
poate genera exceptia intr-un subbloc*/
BEGIN
    SELECT data_finalizare, cod_atelier INTO v_data_fin, v_atelier_vechi FROM
reparatie WHERE cod_reparatie = v_cod_reparatie;
    IF v_data_fin IS NOT NULL THEN
        RAISE reparatie_finalizata;
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista nicio reparatie cu codul
dat ca parametru.');
    WHEN reparatie_finalizata THEN
        RAISE_APPLICATION_ERROR(-20001, 'Reparatia s-a incheiat, nu se mai
poate modifica.');
    END;

    SELECT SUM(cantitate)
    INTO v_total_piese
    FROM lot_componenta WHERE LOWER(numa_componenta) = LOWER(v_nume_piesa) GROUP
BY nume_componenta;

    OPEN calcul_stoc_piesa(v_nume_piesa);
    FETCH calcul_stoc_piesa BULK COLLECT INTO v_stocuri;
    CLOSE calcul_stoc_piesa;

    FOR i IN 1..v_stocuri.last LOOP
        buc_consumate := buc_consumate + v_stocuri(i).consumate;
    END LOOP;

    IF v_total_piese - buc_consumate < v_nr_buc THEN
        RAISE piese_insuficiente;
    ELSE
        afiseaza_mesaj(v_cod_reparatie);
        poz := 1;
        piese_adaugate := 0;
        piese_adaugate_din_lot := 0;
        WHILE piese_adaugate < v_nr_buc LOOP
            nr_ord_piesa := v_stocuri(poz).consumate + piese_adaugate_din_lot +
1;
            --daca am terminat un lot, trec la urmatorul
            IF nr_ord_piesa > v_stocuri(poz).buc_livrare THEN
                poz := poz + 1;
                piese_adaugate_din_lot := 0;
                CONTINUE;
            END IF;
            INSERT INTO COMPONENTA VALUES(v_stocuri(poz).cod_lot, nr_ord_piesa,
v_cod_reparatie);
        END LOOP;
    END IF;
END;

```

```

        piese_adaugate := piese_adaugate + 1;
        piese_adaugate_din_lot := piese_adaugate_din_lot + 1;
    END LOOP;
END IF;

<<mutare_reparatie>>
DECLARE
    total_libere NUMBER;
BEGIN
    total_libere := 0;
    FOR c IN depou_cu_atelier LOOP
        IF c.cod_atelier = v_atelier_vechi THEN
            RAISE deja_in_atelier;
        END IF;
        total_libere := total_libere + c.locuri_libere;
    END LOOP;

    IF total_libere = 0 THEN
        RAISE ateliere_pline;
    END IF;

    --daca nu se arunca exceptiile, inseamna ca se poate realiza mutarea
    OPEN depou_cu_atelier;
    FETCH depou_cu_atelier INTO v_depou, v_atelier_nou, v_capacitate,
v_locuri_libere;

    SELECT i.cod_inginer INTO v_inginer_nou
    FROM angajat a JOIN inginer i ON a.cod_angajat = i.cod_inginer
    WHERE a.cod_depou = v_depou;

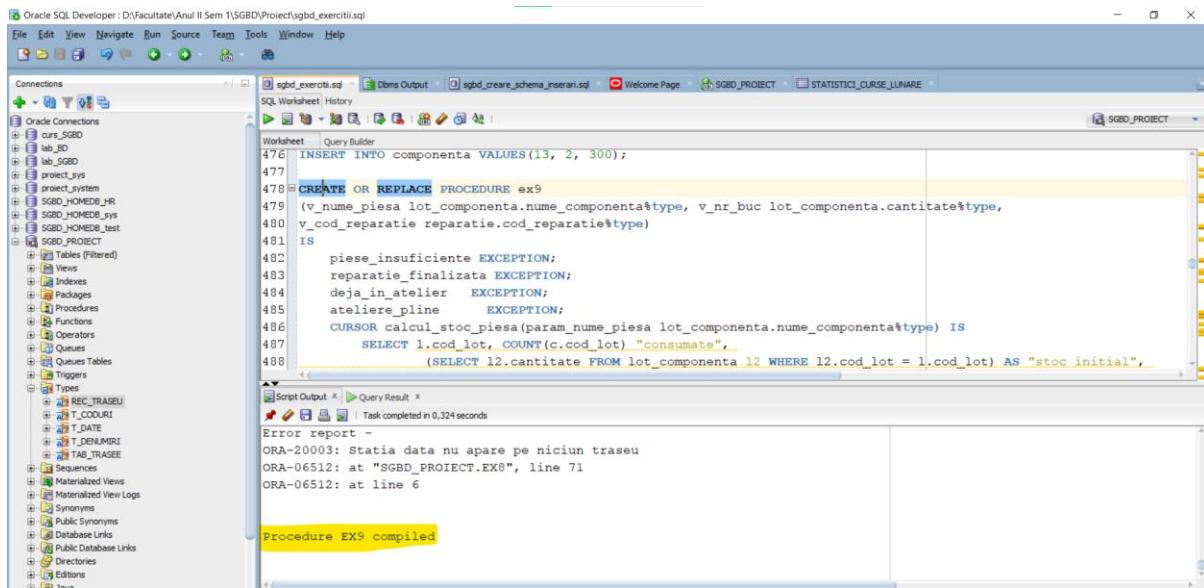
    UPDATE reparatie
    SET cod_atelier = v_atelier_nou,
        cod_inginer = v_inginer_nou
    WHERE cod_reparatie = v_cod_reparatie;
    dbms_output.put_line('S-a produs mutarea de la atelierul ' ||
v_atelier_vechi || ' la atelierul ' || v_atelier_nou);

    CLOSE depou_cu_atelier;
EXCEPTION
    WHEN deja_in_atelier THEN
        dbms_output.put_line('Nu s-a produs mutarea, deoarece reparatia deja
are loc intr-un atelier care se afla la un depou.');
        dbms_output.new_line;
    WHEN ateliere_pline THEN
        dbms_output.put_line('Nu s-a produs mutarea, deoarece toate ateliere
aflate in cadrul depourilor se afla la capacitate maxima.');
        dbms_output.new_line;
    END;

    afiseaza_mesaj(v_cod_reparatie);

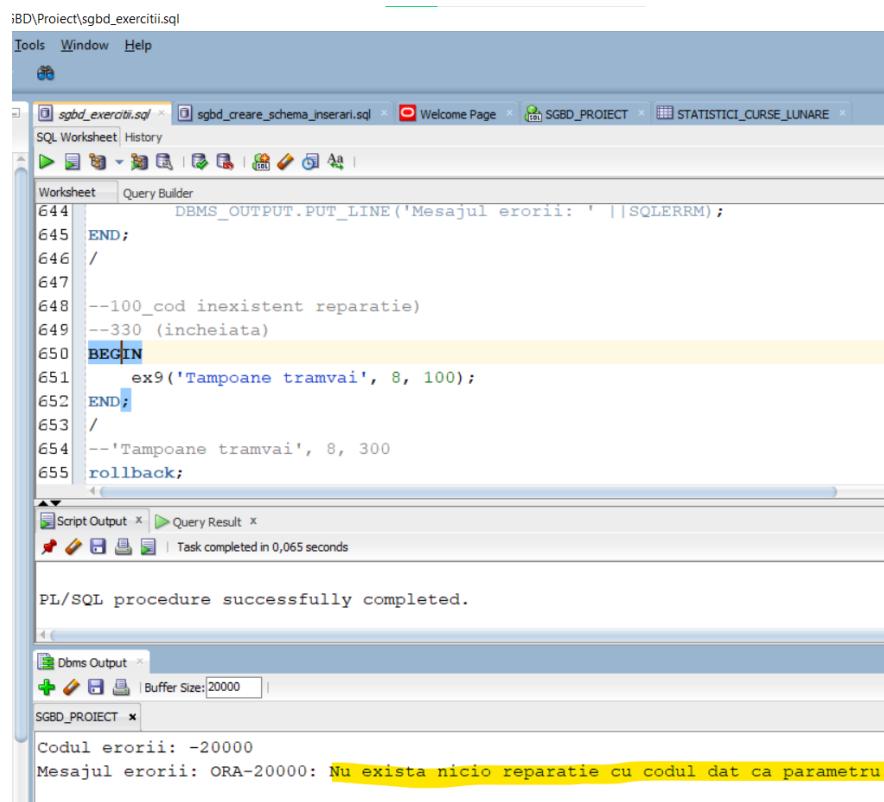
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nu exista nicio piesa cu acest nume.');
    WHEN piese_insuficiente THEN
        RAISE_APPLICATION_ERROR(-20002, 'Nu exista suficiente piese pentru a
putea continua reparatia.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Codul erorii: ' || SQLCODE);
        DBMS_OUTPUT.PUT_LINE('Mesajul erorii: ' || SQLERRM);
END;
/

```



Exemplu 1: no data found pentru cod reparatie

```
BEGIN
    ex9('Tampoane tramvai', 8, 100);
END;
/
```



Exemplu 2: reparatia este deja finalizată

```
BEGIN  
    ex9('Tampoane tramvai', 8, 330);
```

```
END;
/
```

```

648 --100_cod inexistent reparatie)
649 --330 (incheiata)
650 BEGIN
651   ex9('Tampoane tramvai', 8, 330);
652 END;
653 /
654 --'Tampoane tramvai', 8, 300
655 rollback;
656
657 SELECT * FROM lot_componenta;
658 SELECT * FROM componenta order by 1;
659 select * from reparatie order by 3, 6;
660

```

PL/SQL procedure successfully completed.

DBMS Output:

Codul erorii: -20001
Mesajul erorii: ORA-20001: Reparatia s-a incheiat, nu se mai poate modifica.

Exemplu 3: nu sunt suficiente piese în stoc

```

BEGIN
  ex9('Boghiu', 2, 270);
END;
/
```

```

648 --100_cod inexistent reparatie)
649 --330 (incheiata)
650 BEGIN
651   ex9('Boghiu', 2, 270);
652 END;
653 /

```

Error starting at line : 650 in command -

```

BEGIN
  ex9('Boghiu', 2, 270);
END;
Error report -
ORA-20002: Nu exista suficiente piese pentru a putea continua reparatia.
ORA-06512: at "SGBD_PROJECT.EX9", line 164
ORA-06512: at line 2

```

Exemplu 4: no_data_found pentru piesă

```

BEGIN
  ex9('Stergator', 2, 270);
END;
/
```

```

Worksheet | Query Builder
649 --330 (incheiata)
650 BEGIN
651   ex9('Stergator', 2, 270);
652 END;
653 /
654 --'Tampoane tramvai', 8, 300

Script Output | Query Result | Task completed in 0,059 seconds
BEGIN
  ex9('Stergator', 2, 270);
END;
Error report -
ORA-20001: Nu exista nicio piesa cu acest nume.
ORA-06512: at "SGBD_PROIECT.EX9", line 162
ORA-06512: at line 2

```

Exemplu 5: sunt suficiente piese în stoc, mutarea se realizează (cazul ok)

```

BEGIN
  ex9('Tampoane tramvai', 8, 300);
END;
/

```

```

sgbd_exercitii.sql | sgbd_create_schema_inserari.sql | Welcome Page | SGBD_PROJECT | STATISTICI_CURSE_LUNARE |
Worksheet | History | Script Output | Query Result | Task completed in 0,05 seconds | SGBD_PROJECT
649 --330 (incheiata)
650 BEGIN
651   ex9('Tampoane tramvai', 8, 300);
652 END;
653 /
654 --'Tampoane tramvai', 8, 300

PL/SQL procedure successfully completed.

DBMS_OUTPUT
+ Buffer Size: 20000 |
SGBD_PROJECT
Vehiculul V3A-93 cu o vechime de 33 ani, produs de R.A.T.B. URAC se afla in reparatie din data de 29.07.2024 la Atelier Unirii.
Reparatia este coordonata de catre Ing. Ciobanu Luminita.
Pana la acest moment, s-au inlocuit 5 piese.

S-a produs mutarea de la atelierul 200 la atelierul 500.
Vehiculul V3A-93 cu o vechime de 33 ani, produs de R.A.T.B. URAC se afla in reparatie din data de 29.07.2024 la Atelier Vatra Luminoasa.
Reparatia este coordonata de catre Ing. Ciobanu Alexandru.
Pana la acest moment, s-au inlocuit 13 piese.

```

Exemplu 6: sunt suficiente piese în stoc, mutarea NU se realizează (exceptia deja_in_atelier)

```

BEGIN
  ex9('Tampoane tramvai', 8, 270);
END;
/

```

```

649 --330 (incheiata)
650 BEGIN
651     ex9('Tampoane tramvai', 8, 270);
652 END;
653 /
654 --'Tampoane tramvai', 8, 300
4

```

PL/SQL procedure successfully completed.

Vehicul Imperio Metropolitan cu o vechime de 1 ani, produs de Astra Arad se afla in reparatie din data de 25.09.2024 la Atelier Tital.
Reparatia este coordonata de catre Ing. Tudor Marius.
Pana la acest moment, s-au inlocuit 1 piese.

Nu s-a produs mutarea, decarece reparatia deja are loc intr-un atelier care se afla la un depou.

Vehicul Imperio Metropolitan cu o vechime de 1 ani, produs de Astra Arad se afla in reparatie din data de 25.09.2024 la Atelier Tital.
Reparatia este coordonata de catre Ing. Tudor Marius.
Pana la acest moment, s-au inlocuit 5 piese.

Exemplu 7: sunt suficiente piese în stoc, mutarea NU se realizează (excepția ateliere_pline)

```

BEGIN
    ex9('Tampoane tramvai', 1, 290);
    ex9('Tampoane tramvai', 1, 300);
    ex9('Tampoane tramvai', 1, 310);
    ex9('Tampoane tramvai', 1, 320);
    --din acest moment, nu se mai pot realiza transferuri deoarece ateliere sunt
pline
    ex9('Tampoane tramvai', 1, 190);
END;
/

```

Pana la acest moment, s-au inlocuit 3 piese.

S-a produs mutarea de la atelierul 300 la atelierul 500
Vehiculul V3A-93 cu o vechime de 28 ani, produs de R.A.T.B. URAC se afla in reparatie din data de 10.11.2024 la Atelier Vatra Luminoasa.
Reparatia este coordonata de catre Ing. Ciobanu Alexandru.
Pana la acest moment, s-au inlocuit 4 piese.

Vehiculul m5.65 cu o vechime de 65 ani, produs de Rathgeber München se afla in reparatie din data de 22.10.2024 la Atelier Berceni.
Reparatia este coordonata de catre Ing. Dumitrescu Florina.
Pana la acest moment, s-au inlocuit 1 piese.

S-a produs mutarea de la atelierul 300 la atelierul 400
Vehiculul m5.65 cu o vechime de 65 ani, produs de Rathgeber München se afla in reparatie din data de 22.10.2024 la Atelier Tital.
Reparatia este coordonata de catre Ing. Dumitrescu Florina.
Pana la acest moment, s-au inlocuit 2 piese.

Vehiculul Kent C18 cu o vechime de 7 ani, produs de Otokar se afla in reparatie din data de 24.11.2024 la Atelier Berceni.
Reparatia este coordonata de catre Ing. Constantinescu Carmen.
Pana la acest moment, s-au inlocuit 1 piese.

Nu s-a produs mutarea, deoarece toate atelierele aflate in cadrul depourilor se afla la capacitate maxima.

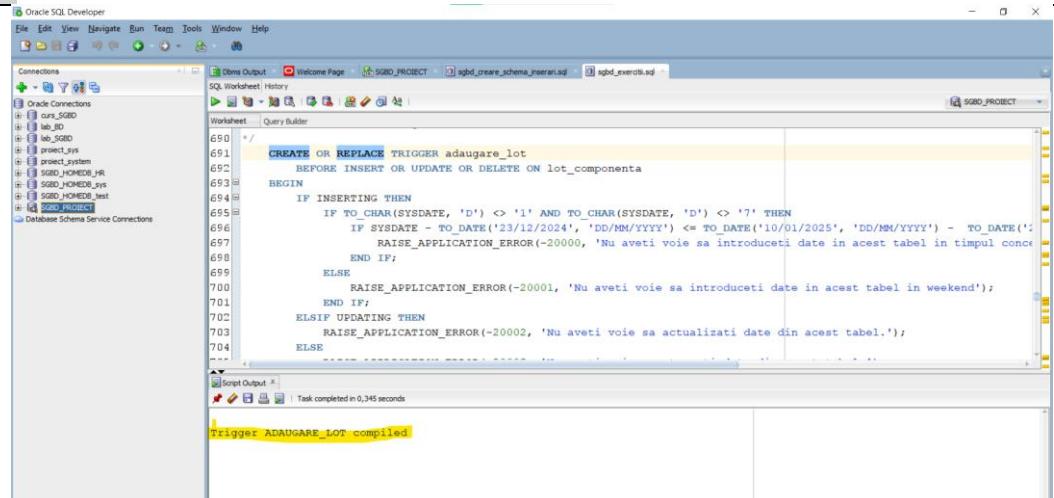
Vehiculul Kent C18 cu o vechime de 7 ani, produs de Otokar se afla in reparatie din data de 24.11.2024 la Atelier Berceni.
Reparatia este coordonata de catre Ing. Constantinescu Carmen.
Pana la acest moment, s-au inlocuit 2 piese.

Cerință 10: Trigger LMD la nivel de comandă

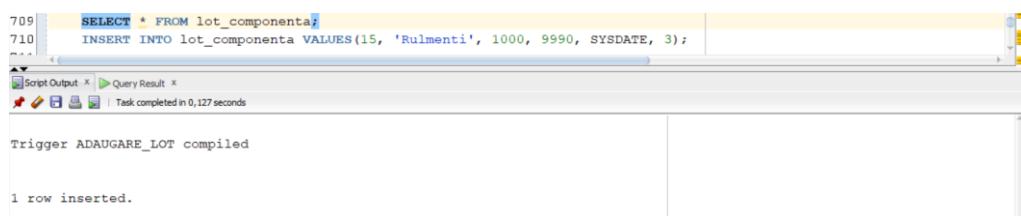
Definiți un *trigger* de tip LMD la nivel de comandă. Declanșați *trigger*-ul.

Am definit un trigger care permite inserarea în tabelul `lot_componentă` doar în zilele lucrătoare (de luni până vineri), **exceptând perioada 23 decembrie 2024 – 10 ianuarie 2025** (considerată perioadă de concediu), și nu permite deloc actualizarea sau ștergerea informațiilor din acea tabelă.

```
CREATE OR REPLACE TRIGGER adaugare_lot
    BEFORE INSERT OR UPDATE OR DELETE ON lot_componenta
BEGIN
    IF INSERTING THEN
        IF TO_CHAR(SYSDATE, 'D') <> '1' AND TO_CHAR(SYSDATE, 'D') <> '7' THEN
            IF SYSDATE - TO_DATE('23/12/2024', 'DD/MM/YYYY') <=
TO_DATE('10/01/2025', 'DD/MM/YYYY') - TO_DATE('23/12/2024', 'DD/MM/YYYY') THEN
                RAISE_APPLICATION_ERROR(-20000, 'Nu aveti voie sa introduceti date in acest tabel in timpul concediului');
            END IF;
        ELSE
            RAISE_APPLICATION_ERROR(-20001, 'Nu aveti voie sa introduceti date in acest tabel in weekend');
        END IF;
    ELSIF UPDATING THEN
        RAISE_APPLICATION_ERROR(-20002, 'Nu aveti voie sa actualizati date din acest tabel.');
    ELSE
        RAISE_APPLICATION_ERROR(-20003, 'Nu aveti voie sa stergeti date din acest tabel.');
    END IF;
END;
/
```



```
INSERT INTO lot_componenta VALUES(15, 'Rulmenti', 1000, 9990, SYSDATE, 3);
```



```
708 /
709 SELECT * FROM lot_componenta;
710 INSERT INTO lot_componenta VALUES(15, 'Rulmenti', 1000, 9990, SYSDATE, 3);
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
```

În acest caz, se declanșează triggerul, însă nu are niciun efect vizibil, deoarece a fost permisă inserarea în condițiile în care s-a încheiat perioada de concediu (inserarea a fost făcută în ziua de vineri, 10 ianuarie).

Declanșarea (+ efectul):

```
DELETE FROM lot_componenta WHERE lower(numa_componenta) LIKE '%autobuz%';
```

```
715  
716     DELETE FROM lot_componenta WHERE lower(nume_componenta) LIKE '%autobuz%';  
717  
ORA-04088: error during execution of trigger 'SGBD_PROIECT.ADAUGARE_LOT'  
  
Error starting at line : 716 in command -  
    DELETE FROM lot_componenta WHERE lower(nume_componenta) LIKE '%autobuz%'  
Error at Command Line : 716 Column : 17  
Error report -  
SQL Error: ORA-20003: Nu aveti voie sa stergeti date din acest tabel,  
ORA-06512: at "SGBD_PROIECT.ADAUGARE_LOT", line 13  
ORA-04088: error during execution of trigger 'SGBD_PROIECT.ADAUGARE_LOT'
```

Cerință 11: Trigger LMD la nivel de linie

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

Am creat un trigger care să se declanșeze când se încearcă o modificare asupra tabelei **cursă**:

- să se permită introducerea unei curse, doar dacă șoferul va conduce un vehicul garat în depoul în care este repartizat și el, pe un traseu corespunzător (ex: să nu conducă un autobuz pe un traseu destinat troleibuzelor) și doar dacă vehiculul respectă categoria de mijloc de transport de pe permisul șoferului;
- pentru o cursă deja existentă, nu se pot actualiza decât codul vehiculului de transport sau numărului traseului (un update care ar încălca condițiile de mai sus nu ar trebui permis);
- după orice modificare (insert/delete), să se actualizeze tabela **statistici_curse_lunare**, în care, pentru fiecare lună, se salvează numărul de curse efectuate cu autobuzul, troleibuzul și tramvaiul: după o inserare, crește cu 1 valoarea coloanei corespunzătoare numărului de curse efectuate cu categoria de mijloc de transport din care face parte vehiculul din cursa inserată, iar după o ștergere, valoarea coloanei scade cu 1.

```
--definirea tabelei cu statistici și inserarea datelor
CREATE TABLE statistici_curse_lunare(
    cod_luna          VARCHAR2(8)      CONSTRAINT pk_stats_curse PRIMARY KEY,
    nr_curse_bus     NUMBER,
    nr_curse_trol    NUMBER,
    nr_curse_tram   NUMBER
);
--DROP TABLE statistici_curse_lunare;
DECLARE
    var1  VARCHAR2(8);
    var2  NUMBER;
    var3  NUMBER;
    var4  NUMBER;
    CURSOR curse_lunare IS
        SELECT TO_CHAR(c1.data_cursa, 'MM-YYYY') "luna" ,
               (SELECT COUNT(c2.cod_cursa) FROM cursa c2 JOIN traseu t ON
c2.numar_traseu = t.numar_traseu
                  WHERE TO_CHAR(c1.data_cursa, 'MM-YYYY') = TO_CHAR(c2.data_cursa,
'MM-YYYY') AND t.vehicule_folosite = 'autobuz') "curse_autobuz",
               (SELECT COUNT(c2.cod_cursa) FROM cursa c2 JOIN traseu t ON
c2.numar_traseu = t.numar_traseu
                  WHERE TO_CHAR(c1.data_cursa, 'MM-YYYY') = TO_CHAR(c2.data_cursa,
'MM-YYYY') AND t.vehicule_folosite = 'troleibuz') "curse_troleibuz",
               (SELECT COUNT(c2.cod_cursa) FROM cursa c2 JOIN traseu t ON
c2.numar_traseu = t.numar_traseu
                  WHERE TO_CHAR(c1.data_cursa, 'MM-YYYY') = TO_CHAR(c2.data_cursa,
'MM-YYYY') AND t.vehicule_folosite = 'tramvai') "curse_tramvai"
        FROM cursa c1
        GROUP BY TO_CHAR(c1.data_cursa, 'MM-YYYY')
        ORDER BY 1;
BEGIN
    OPEN curse_lunare;
    LOOP
        FETCH curse_lunare INTO var1, var2, var3, var4;
        EXIT WHEN curse_lunare%NOTFOUND;
        INSERT INTO statistici_curse_lunare VALUES(var1, var2, var3, var4);
    END LOOP;
    CLOSE curse_lunare;
END;
/
```

```
--SELECT * FROM statistici_curse_lunare;
```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the path is D:\Facultate\Anul II Sem 1\SGBD\Proiect\sgbd_exercitii.sql. The main area displays the following SQL code:

```
753 SELECT * FROM statistici_curse_lunare;
754
755 /*
```

Below the code, the "Query Result" tab is selected, showing the output of the query:

COD_LUNA	NR_CURSE_BUS	NR_CURSE_TROL	NR_CURSE_TRAM
1 01-2023	3	3	3
2 01-2024	3	8	2
3 01-2025	0	3	0
4 02-2023	6	3	5
5 02-2024	3	4	1
6 03-2023	5	1	2
7 03-2024	2	6	7
8 04-2023	1	6	5
9 04-2024	6	2	5
10 05-2023	1	8	5
11 05-2024	5	4	5
12 06-2023	4	0	5
13 06-2024	2	2	2
14 07-2023	2	5	4
15 07-2024	2	6	3
16 08-2023	3	3	4
17 08-2024	3	5	5
18 09-2023	8	7	7
19 09-2024	6	1	4
20 10-2023	3	7	10
21 10-2024	2	7	3

Am implementat o funcție care, pentru un cod de șofer, un traseu și un cod de vehicul, verifică dacă șoferul poate fi repartizat pe acel traseu, cu acel vehicul:

```
CREATE OR REPLACE FUNCTION repartitie_corecta
(p_cod_sofer sofer.cod_sofer%type, p_nr_traseu traseu.numar_traseu%type,
p_cod_vehicul vehicul.cod_vehicul%type)
RETURN BOOLEAN
IS
    permite BOOLEAN;
    v_categ_permis      sofer.categorie_permis%type;
    v_cod_depou_ang     angajat.cod_depou%type;
    v_cod_depou_veh     angajat.cod_depou%type;
    v_tip_traseu        traseu.vhicule_folosite%type;
    v_tip_vehicul       model.tip_mijloc_transport%type;

BEGIN
    BEGIN
        SELECT s.categorie_permis, a.cod_depou INTO v_categ_permis,
v_cod_depou_ang
        FROM sofer s JOIN angajat a ON s.cod_sofer = a.cod_angajat
        WHERE s.cod_sofer = p_cod_sofer;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Cod sofer inexistent');
    END;

    BEGIN
        SELECT veh.cod_depou, m.tip_mijloc_transport INTO v_cod_depou_veh,
v_tip_vehicul
        FROM vehicul veh JOIN model m ON m.cod_model = veh.cod_model
        WHERE veh.cod_vehicul = p_cod_vehicul;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Vehicul inexistent');
    END;

    BEGIN

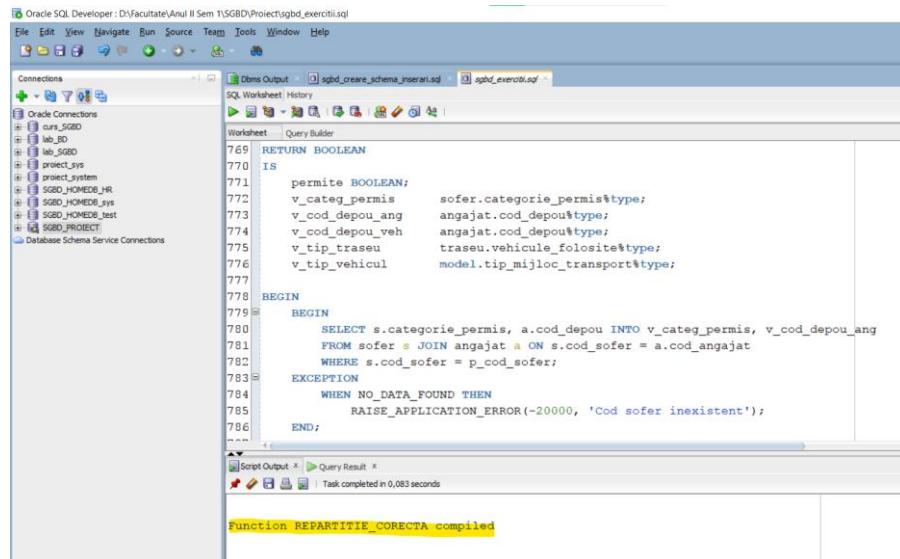
```

```

        SELECT vehicule_folosite INTO v_tip_traseu FROM traseu WHERE numar_traseu
= p_nr_traseu;
      EXCEPTION
        WHEN NO_DATA_FOUND THEN
          RAISE_APPLICATION_ERROR(-20000, 'Traseu inexistent');
    END;

  permite := FALSE;
  IF v_cod_depou_veh = v_cod_depou_ang THEN
    IF v_tip_traseu = v_tip_vehicul THEN
      CASE v_categ_permis
        WHEN 'BUS' THEN
          IF v_tip_traseu = 'autobuz' THEN
            permite := TRUE;
          END IF;
        WHEN 'TRL' THEN
          IF v_tip_traseu = 'troleibuz' THEN
            permite := TRUE;
          END IF;
        ELSE
          IF v_tip_traseu = 'tramvai' THEN
            permite := TRUE;
          END IF;
        END CASE;
      END IF;
    END IF;
    RETURN permite;
  END repartitie_corecta;
/

```



Triggerul:

```

CREATE OR REPLACE TRIGGER modif_cursa
  BEFORE INSERT OR UPDATE OR DELETE ON cursa
  FOR EACH ROW
DECLARE
  e_permis BOOLEAN;
  tip_cursa  traseu.vehicule_folosite%type;
  v_data    VARCHAR2(8);
  v_nr      NUMBER;
BEGIN
  IF INSERTING THEN

```

```

    e_permis := repartitie_corecta(:NEW.cod_sofer, :NEW.numar_traseu,
:NEW.cod_vehicul_tr);
    IF e_permis = FALSE THEN
        RAISE_APPLICATION_ERROR(-20001, 'Datele nu respecta conditiile:
verificati daca soferul si vehiculul apartin de acelasi depou, sau daca traseul a
fost ales corect');
    ELSE
        SELECT vehicule_folosite INTO tip_cursa FROM traseu WHERE
numar_traseu = :NEW.numar_traseu;
        v_data := TO_CHAR(:NEW.data_cursa, 'MM-YYYY');
        SELECT COUNT(*) INTO v_nr FROM statistici_curse_lunare WHERE cod_luna
= v_data;
        --daca pana acum nu au mai existat curse in acea luna:
        IF v_nr = 0 THEN
            INSERT INTO statistici_curse_lunare VALUES(v_data, 0, 0, 0);
        END IF;
        CASE tip_cursa
            WHEN 'autobuz' THEN
                UPDATE statistici_curse_lunare
                SET nr_curse_bus = nr_curse_bus + 1
                WHERE cod_luna = v_data;
            WHEN 'troleibuz' THEN
                UPDATE statistici_curse_lunare
                SET nr_curse_trol = nr_curse_trol + 1
                WHERE cod_luna = v_data;
            ELSE
                UPDATE statistici_curse_lunare
                SET nr_curse_tram = nr_curse_tram + 1
                WHERE cod_luna = v_data;
        END CASE;
    END IF;
ELSIF UPDATING('numar_traseu') OR UPDATING('cod_vehicul_tr') THEN
    e_permis := repartitie_corecta(:NEW.cod_sofer, :NEW.numar_traseu,
:NEW.cod_vehicul_tr);
    IF e_permis = FALSE THEN
        RAISE_APPLICATION_ERROR(-20001, 'Modificarea nu este corecta:
verificati daca soferul si vehiculul apartin de acelasi depou, sau daca traseul a
fost ales corect');
    END IF;
ELSIF DELETING THEN
    SELECT vehicule_folosite INTO tip_cursa FROM traseu WHERE numar_traseu =
:OLD.numar_traseu;
    v_data := TO_CHAR(:OLD.data_cursa, 'MM-YYYY');
    CASE tip_cursa
        WHEN 'autobuz' THEN
            UPDATE statistici_curse_lunare
            SET nr_curse_bus = nr_curse_bus - 1
            WHERE cod_luna = v_data;
        WHEN 'troleibuz' THEN
            UPDATE statistici_curse_lunare
            SET nr_curse_trol = nr_curse_trol - 1
            WHERE cod_luna = v_data;
        ELSE
            UPDATE statistici_curse_lunare
            SET nr_curse_tram = nr_curse_tram - 1
            WHERE cod_luna = v_data;
    END CASE;
ELSE
    RAISE_APPLICATION_ERROR(-20002, 'Se pot actualiza doar linia sau
vehiculul.');
END IF;
END;
/

```

```

CREATE OR REPLACE TRIGGER modif_cursa
  BEFORE INSERT OR UPDATE OR DELETE ON cursa
  FOR EACH ROW
DECLARE
  e_permis BOOLEAN;
  tip_cursa  traseu.vehicule_folosite%type;
  v_data  VARCHAR2(8);
  v_nr   NUMBER;
BEGIN
  IF INSERTING THEN
    e_permis := repartitie_corecta(:NEW.cod_sofer, :NEW.numar_traseu, :NEW.cod_vehicul_tr);
    IF e_permis = FALSE THEN
      RAISE_APPLICATION_ERROR(-20001, 'Datele nu respectă condițiile: verificati daca soferul si vehiculul apar');
    ELSE
      SELECT vehicule_folosite INTO tip_cursa FROM traseu WHERE numar_traseu = :NEW.numar_traseu;
      v_data := TO_CHAR(:NEW.data_cursa, 'MM-YYYY');
      SELECT COUNT(*) INTO v_nr FROM statistici_curse_lunare WHERE cod_luna = v_data;
      IF v_nr = 0 THEN
        ...
      END IF;
    END IF;
  END IF;
END;
  
```

Trigger MODIF_CURSE compiled

Exemplu 1: inserare permisă (modificările sunt vizibile în tabela statistici_curse_lunare)

```

INSERT INTO CURSA VALUES(2000, 119, 1000, 182, SYSDATE, TO_DATE('05:43',
'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'),
TO_DATE('22:08', 'HH24:MI'), 17);
  
```

```

898 INSERT INTO CURSA VALUES(2000, 119, 1000, 182, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:08', 'HH24:MI'), 17);
899 INSERT INTO CURSA VALUES(2000, 119, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:08', 'HH24:MI'), 17);
900 --119, 182, 1000
  
```

1 row inserted.

```

899 INSERT INTO CURSA VALUES(2000, 119, 1000, 14, SYSDATE)
--119, 182, 1000
901 UPDATE CURSA SET numar_ture = 2 WHERE cod_cursa = 101
902 UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa =
  
```

COD_LUNA	NR_CURSE_BUS	NR_CURSE_TROL	NR_CURSE_TRAM
1 01-2023	3	3	3
2 01-2024	3	8	2
3 01-2025	1	0	0
4 02-2023	6	3	5
5 02-2024	3	4	1
6 03-2023	5	1	2
7 03-2024	2	6	7
8 04-2023	1	6	5
9 04-2024	6	2	5
10 05-2023	1	8	5
11 05-2024	5	4	5

După inserare, a apărut o cursă de autobuz în luna ianuarie 2025

Exemplu 2: inserarea nu este permisă

2.1. se respectă constrângerile referențiale (există șoferul, traseul, vehiculul), dar nu se respectă condițiile impuse (1000 este autobuz, traseul 14 este o linie de tramvai)

```
INSERT INTO CURSA VALUES(2000, 119, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'),  
TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:08',  
'HH24:MI'), 17);
```

```
898 INSERT INTO CURSA VALUES(2000, 119, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:08', 'HH24:MI'), 17);  
899 INSERT INTO CURSA VALUES(2000, 119, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:08', 'HH24:MI'), 17);  
900 UPDATE CURSA SET numar_ture = 2 WHERE cod_cursa = 1000; --nepermis  
901 UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa = 1000; --ok  
902  
or starting at line : 899 in command -  
INSERT INTO CURSA VALUES(2000, 119, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:  
or at Command Line : 899 Column : 13  
or report -  
Error: ORA-20001: Datele nu respectă condițiile: verificati daca soferul si vehiculul aparțin de același depou, sau daca traseul a fost ales corect  
-06512: at "SGBD_PROJECT.MODIF_CURSE", line 10  
-04088: error during execution of trigger 'SGBD_PROJECT.MODIF_CURSE'
```

2.2. nu există șofer / traseu / vehicul cu codul dat: în aceste cazuri, se vor arunca excepțiile tratate în funcția care verifică respectarea condițiilor

```
INSERT INTO CURSA VALUES(2000, 1, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'),  
TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:08',  
'HH24:MI'), 17);
```

```
899 INSERT INTO CURSA VALUES(2000, 1, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:  
--119, 182, 1000  
900 UPDATE CURSA SET numar_ture = 2 WHERE cod_cursa = 1000; --nepermis  
901 UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa = 1000; --ok  
902 UPDATE CURSA SET cod_vehicul_tr = 9096 WHERE cod_cursa = 1000; --modificare gresita  
903  
or starting at line : 899 in command -  
INSERT INTO CURSA VALUES(2000, 1, 1000, 14, SYSDATE, TO_DATE('05:43', 'HH24:MI'), TO_DATE('06:15', 'HH24:MI'), TO_DATE('21:59', 'HH24:MI'), TO_DATE('22:  
Error at Command Line : 899 Column : 13  
Error report -  
SQL Error: ORA-20000: Cod sofer inexistent  
ORA-06512: at "SGBD_PROJECT.REPARTITIE_CORECTA", line 19  
ORA-06512: at "SGBD_PROJECT.MODIF_CURSE", line 8  
ORA-04088: error during execution of trigger 'SGBD_PROJECT.MODIF_CURSE'  
20000. 00000 - "%"  
*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 3: actualizare permisă

```
UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa = 1000;
```

```
902 UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa = 1000; --ok  
903 UPDATE CURSA SET cod_vehicul_tr = 9096 WHERE cod_cursa = 1000; --modificare gresita  
904 DELETE FROM cursa WHERE cod_cursa = 1000; -- -1 la tramvaie pe ianuarie 2023  
905  
or starting at line : 902 in command -  
UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa = 1000; --ok  
1  
1 row updated.
```

Înainte:

```
905 | SELECT * FROM cursa WHERE cod_cursa = 1000;
906 | 
```

COD_CURSA	COD_SOFER	COD_VEHICUL_TR	NUMAR_TRASEU	DATA_CURSA	ORA_PRELUCARE	ORA_INCEPERE_TURE	ORA_FINALIZARE_TURE	ORA_PREDARE	NUMAR_TURE
1	1000	127	9094	17-JAN-23	01-JAN-25	01-JAN-25	01-JAN-25	01-JAN-25	23

După:

```
906 | SELECT * FROM cursa WHERE cod_cursa = 1000;
907 | 
```

COD_CURSA	COD_SOFER	COD_VEHICUL_TR	NUMAR_TRASEU	DATA_CURSA	ORA_PRELUCARE	ORA_INCEPERE_TURE	ORA_FINALIZARE_TURE	ORA_PREDARE	NUMAR_TURE
1	1000	127	9094	17-JAN-23	01-JAN-25	01-JAN-25	01-JAN-25	01-JAN-25	23

Exemplu 4: actualizări nepermise

4.1. se încearcă modificarea altei coloane, în afară de număr_traseu sau cod_vehicul:

```
UPDATE CURSA SET numar_ture = 2 WHERE cod_cursa = 1000;
```

```
901 | UPDATE CURSA SET numar_ture = 2 WHERE cod_cursa = 1000; --nepermis
902 | UPDATE CURSA SET numar_traseu = 14 WHERE cod_cursa = 1000; --ok
903 | UPDATE CURSA SET cod vehicul tr = 9096 WHERE cod cursa = 1000; --modificare gresita
```

COD_CURSA	COD_SOFER	COD_VEHICUL_TR	NUMAR_TRASEU	DATA_CURSA	ORA_PRELUCARE	ORA_INCEPERE_TURE	ORA_FINALIZARE_TURE	ORA_PREDARE	NUMAR_TURE
1	1000	127	9094	17-JAN-23	01-JAN-25	01-JAN-25	01-JAN-25	01-JAN-25	23

Error starting at line : 901 in command -
 UPDATE CURSA SET numar_ture = 2 WHERE cod_cursa = 1000
 Error at Command Line : 901 Column : 8
 Error report -
 SQL Error: ORA-20002: Se pot actualiza doar linia sau vehiculul.
 ORA-06512: at "SGBD_PROIECT.MODIF_CURSE", line 56
 ORA-04088: error during execution of trigger 'SGBD_PROIECT.MODIF_CURSE'

4.2. modificarea nu mai respectă condițiile modelului (vehiculul 9096 este garat la alt depou):

```
UPDATE CURSA SET cod_vehicul_tr = 9096 WHERE cod_cursa = 1000;
```

```
903 | UPDATE CURSA SET cod_vehicul_tr = 9096 WHERE cod_cursa = 1000; --modificare gresita
904 | 
```

COD_CURSA	COD_SOFER	COD_VEHICUL_TR	NUMAR_TRASEU	DATA_CURSA	ORA_PRELUCARE	ORA_INCEPERE_TURE	ORA_FINALIZARE_TURE	ORA_PREDARE	NUMAR_TURE
1	1000	127	9094	17-JAN-23	01-JAN-25	01-JAN-25	01-JAN-25	01-JAN-25	23

Error starting at line : 903 in command -
 UPDATE CURSA SET cod_vehicul_tr = 9096 WHERE cod_cursa = 1000
 Error at Command Line : 903 Column : 8
 Error report -
 SQL Error: ORA-20001: Modificarea nu este corecta: verificati daca soferul si vehiculul aparțin de același depou, sau daca traseul a fost ales corect
 ORA-06512: at "SGBD_PROIECT.MODIF_CURSE", line 36
 ORA-04088: error during execution of trigger 'SGBD_PROIECT.MODIF_CURSE'

Exemplu 5: ștergeri

```
DELETE FROM cursa WHERE TO_CHAR(data_cursa, 'MM-YYYY') = '01-2023';
```

The screenshot shows a SQL query window with the following content:

```
905 | DELETE FROM cursa WHERE TO_CHAR(data_cursa, 'MM-YYYY') = '01-2023';
906 |
```

Below the query, the output pane shows:

Script Output x Query Result x
Task completed in 0,069 seconds

Rollback complete.

9 rows deleted.

După ștergere, nu mai apare nicio cursă în dreptul lunii ianuarie 2023 în tabela cu statistici:

The screenshot shows a SQL query window with the following content:

```
891 | SELECT * FROM statistici_curse_lunare;
```

Below the query, the output pane shows:

Script Output x Query Result x
SQL | All Rows Fetched: 25 in 0,005 seconds

COD_LUNA	NR_CURSE_BUS	NR_CURSE_TROL	NR_CURSE_TRAM
01-2023	0	0	0

Cerință 12: Trigger LDD

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

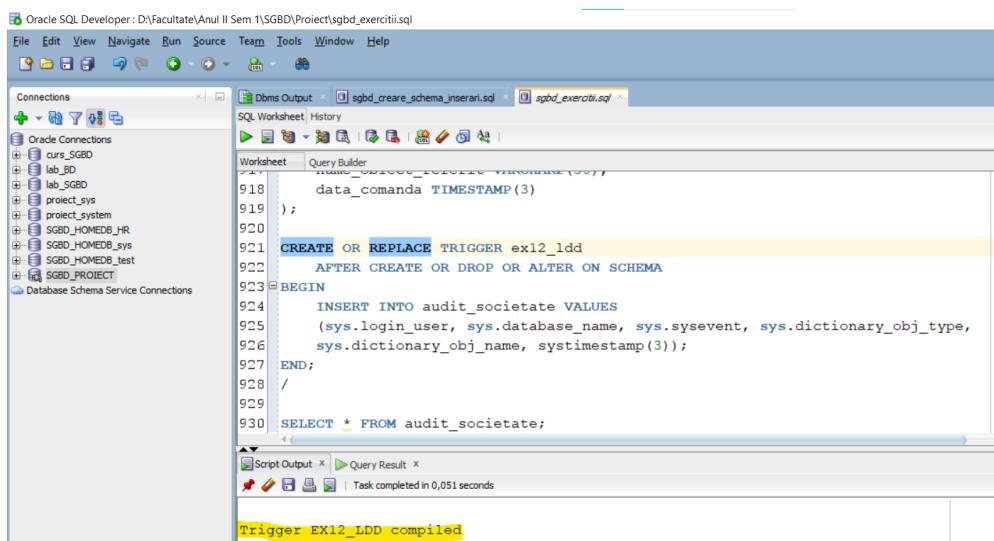
Am definit un trigger care se declanșează de fiecare dată când se execută o operație LDD asupra schemei și inserează într-o tabelă de audit informații referitoare la comandă.

```

CREATE TABLE audit_societate(
    utilizator VARCHAR2(30),
    nume_bd VARCHAR2(30),
    eveniment VARCHAR2(20),
    tip_object_referit VARCHAR2(30),
    nume_object_referit VARCHAR2(30),
    data_comanda TIMESTAMP(3)
);

CREATE OR REPLACE TRIGGER ex12_ldd
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO audit_societate VALUES
        (sys.login_user, sys.database_name, sys.sysevent, sys.dictionary_obj_type,
        sys.dictionary_obj_name, systimestamp(3));
END;
/

```



```

--secvența de comenzi LDD
CREATE OR REPLACE TYPE t_modele IS TABLE OF VARCHAR2(64);
/

CREATE OR REPLACE VIEW viz_vehicule AS
    SELECT v.cod_vehicul, v.producator, m.denumire, m.tip_mijloc_transport,
    TO_CHAR(v.an_achizitie, 'yyyy') "an_achiz"
        FROM vehicul v JOIN model m ON v.cod_model = m.cod_model;

DROP TYPE t_modele;

DROP VIEW viz_vehicule;

```

```

CREATE OR REPLACE VIEW viz_vehicule AS
SELECT v.cod_vehicul, v.producator, m.denumire, m.tip_mijloc_transport, TO_CHAR(v.an_achizitie, 'yyyy') "an_achizitie"
FROM vehicul v JOIN model m ON v.cod_model = m.cod_model;

```

Type T_MODELE compiled

View VIZ_VEHICULE created.

Type T_MODELE dropped.

View VIZ_VEHICULE dropped.

```
SELECT * FROM audit_societate;
```

```

CREATE OR REPLACE VIEW viz_vehicule AS
SELECT v.cod_vehicul, v.producator, m.denumire, m.tip_mijloc_transport, TO_CHAR(v.an_achizitie, 'yyyy') "an_achizitie"
FROM vehicul v JOIN model m ON v.cod_model = m.cod_model;

```

```

DROP TYPE t_modele;
DROP VIEW viz_vehicule;

```

```

SELECT * FROM audit_societate;
--TRUNCATE TABLE audit_societate;

```

UTILIZATOR	NUME_BD	EVENIMENT	TIP_OBJECT_REFERIT	NUME_OBJECT_REFERIT	DATA_COMANDA
1 SGBD_PROJECT ORCL.LOCAL CREATE TYPE			T_MODELE		10-JAN-25 12.03.41.581000000 PM
2 SGBD_PROJECT ORCL.LOCAL CREATE VIEW			VIZ_VEHICULE		10-JAN-25 12.03.43.441000000 PM
3 SGBD_PROJECT ORCL.LOCAL DROP TYPE			T_MODELE		10-JAN-25 12.03.46.049000000 PM
4 SGBD_PROJECT ORCL.LOCAL DROP VIEW			VIZ_VEHICULE		10-JAN-25 12.03.48.157000000 PM

Cerință 13: Pachet

Formulați în limbaj natural o problemă pe care să o rezolvați folosind 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).

Enunț: pentru un cod de reparație și o dată, să se încheie reparația aflată în desfășurare, efectuându-se și următoarele modificări necesare pentru consistența datelor:

- să se actualizeze înregistrarea corespunzătoare din tabela **reparație**: **dată_finalizare** va lua valoarea datei furnizate ca parametru;
- să se insereze în tabela **observație** un mesaj care indică faptul că vehiculul a trecut printr-o reparație. În funcție de numărul de componente înlocuite, reparația se poate clasifica în: minoră (1-4 componente), majoră(5-8 componente), capitală(> 8 componente);
- să se actualizeze starea și data ultimei verificări în tabela **vehicul**, pentru vehiculul care tocmai și-a finalizat reparația
- să se afișeze statistici despre reparație: numărul componentelor folosite în funcție de lotul de proveniență, denumirea componenței, furnizorul lotului, numărul și data livrării
- să se afișeze toate observațiile referitoare la vehicul.

```

CREATE OR REPLACE PACKAGE pack_ex13 IS
    PROCEDURE incheie_reparatie(p_cod_reparatie reparatie.cod_reparatie%type,
                                p_data_final     reparatie.data_finalizare%type);
END;
/


CREATE OR REPLACE PACKAGE BODY pack_ex13 IS
    TYPE tab_obs  IS TABLE OF observatie%rowtype;

    CURSOR c(p_cod_reparatie reparatie.cod_reparatie%type) IS
        SELECT r.cod_reparatie, lc.cod_lot || '/' || TO_CHAR(lc.data_livrare,
'DD.MM.YYYY') lot_data, COUNT(cmp.numar_componenta) nr_comp_folosite,
lc.nume_componenta, f.nume furnizor
        FROM reparatie r
            JOIN componenta cmp ON cmp.cod_reparatie = r.cod_reparatie
            JOIN lot_componenta lc ON lc.cod_lot = cmp.cod_lot
            JOIN furnizor f ON f.cod_furnizor = lc.cod_furnizor
        WHERE r.cod_reparatie = p_cod_reparatie
        GROUP BY r.cod_reparatie, lc.cod_lot || '/' || TO_CHAR(lc.data_livrare,
'DD.MM.YYYY'), lc.nume_componenta, f.nume
        ORDER BY 4;

    TYPE tab_piese IS TABLE OF c%rowtype;

    FUNCTION get_cod_vehicul(p_cod_reparatie    reparatie.cod_reparatie%type)
        RETURN NUMBER IS
        v_cod NUMBER;
    BEGIN
        SELECT cod_vehicul_tr INTO v_cod FROM reparatie WHERE cod_reparatie =
p_cod_reparatie;
        RETURN v_cod;
    END;

```

```

FUNCTION clasifica_reparatie(p_cod_reparatie      reparatie.cod_reparatie%type)
    RETURN VARCHAR2 IS
    v_tip          VARCHAR2(64);
    nr_piese       NUMBER;
BEGIN
    SELECT COUNT(*) INTO nr_piese FROM componenta WHERE cod_reparatie =
p_cod_reparatie;
    CASE
        WHEN nr_piese BETWEEN 1 AND 4 THEN
            v_tip := 'Reparatie minora';
        WHEN nr_piese BETWEEN 5 AND 8 THEN
            v_tip := 'Reparatie majora';
        ELSE
            v_tip := 'Reparatie capitala';
    END CASE;
    RETURN v_tip;
END;

PROCEDURE actualizeaza_vehicul(p_cod_vehicul vehicul.cod_vehicul%type,
                                 p_data_verif  vehicul.data_verificare%type) IS
BEGIN
    UPDATE vehicul
    SET data_verificare = p_data_verif,
        stare = 'functional'
    WHERE cod_vehicul = p_cod_vehicul;
    dbms_output.put_line('Informatiile despre vehicul au fost actualizate.');
END;

PROCEDURE afis_observatii(p_cod_vehicul  observatie.cod_vehicul%type) IS
    v_obs    tab_obs := tab_obs();
BEGIN
    SELECT * BULK COLLECT INTO v_obs FROM observatie WHERE cod_vehicul =
p_cod_vehicul;
    FOR i IN 1..v_obs.last LOOP
        dbms_output.put_line(v_obs(i).descriere || ', adaugata in data de: '
|| TO_CHAR(v_obs(i).data_obs, 'DD.MM.YYYY'));
    END LOOP;
END;

PROCEDURE observatie_noua(p_descriere   observatie.descriere%type,
                           p_data        observatie.data_obs%type,
                           p_cod_vehicul vehicul.cod_vehicul%type) IS
    cod_obs NUMBER;
BEGIN
    SELECT MAX(cod_observatie) + 1 INTO cod_obs FROM observatie;
    INSERT INTO observatie VALUES(cod_obs, p_descriere, p_data,
p_cod_vehicul);
    dbms_output.put_line('Istoricul vehiculului a fost actualizat.');
    dbms_output.put_line('-----');
    dbms_output.put_line('Toate observatiile:');
    afis_observatii(p_cod_vehicul);
    dbms_output.put_line('-----');
END;

PROCEDURE statistici_reparatie(p_cod_reparatie
reparatie.cod_reparatie%type) IS
    v_piese  tab_piese := tab_piese();
BEGIN
    dbms_output.put_line('S-au folosit urmatoarele componente:');
    OPEN c(p_cod_reparatie);
    FETCH c BULK COLLECT INTO v_piese;
    CLOSE c;
    FOR i IN 1..v_piese.last LOOP

```

```

        dbms_output.put_line(v_piese(i).nr_comp_folosite || ' x ' ||
v_piese(i).nume_componenta || ' furnizate de ' || v_piese(i).furnizor || ' in
totul ' || v_piese(i).lot_data);
    END LOOP;
    dbms_output.put_line('-----');
END;

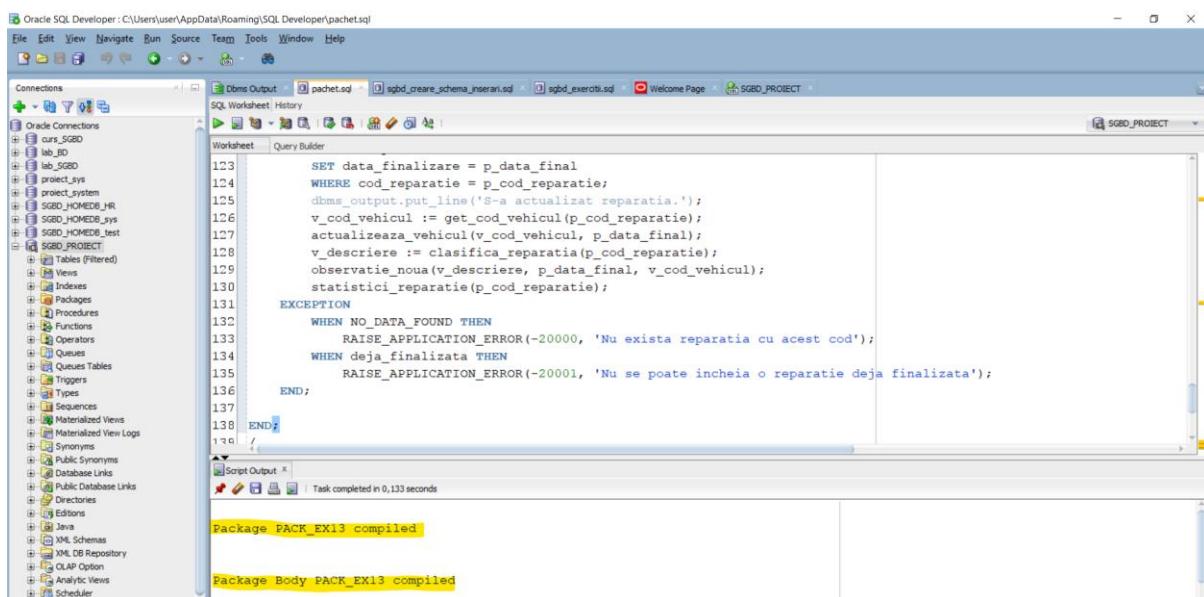
PROCEDURE incheie_reparatie(p_cod_reparatie reparatie.cod_reparatie%type,
                             p_data_final     reparatie.data_finalizare%type)
IS
    v_cod_vehicul      reparatie.cod_vehicul_tr%type;
    v_data_final       reparatie.data_finalizare%type;
    v_descriere        VARCHAR2(64);
    deja_finalizata    EXCEPTION;
BEGIN
    SELECT cod_vehicul_tr, data_finalizare INTO v_cod_vehicul, v_data_final
    FROM reparatie WHERE cod_reparatie = p_cod_reparatie;
    IF v_data_final IS NOT NULL THEN
        RAISE deja_finalizata;
    END IF;

    UPDATE reparatie
    SET data_finalizare = p_data_final
    WHERE cod_reparatie = p_cod_reparatie;

    dbms_output.put_line('S-a actualizat reparatia.');
    v_cod_vehicul := get_cod_vehicul(p_cod_reparatie);
    actualizeaza_vehicul(v_cod_vehicul, p_data_final);
    v_descriere := clasifica_reparatie(p_cod_reparatie);
    observatie_noua(v_descriere, p_data_final, v_cod_vehicul);
    statistici_reparatie(p_cod_reparatie);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista reparatia cu acest cod');
    WHEN deja_finalizata THEN
        RAISE_APPLICATION_ERROR(-20001, 'Nu se poate incheia o reparatie deja
finalizata');
    END;
END;
/

```



Utilizarea pachetului:

```
EXECUTE pack_ex13.incheie_reparatia(300, SYSDATE);
```

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, the 'pachet.sql' tab is active. The main area displays the following PL/SQL code:

```
139 /  
140 rollback;  
141 EXECUTE pack_ex13.incheie_reparatia(300, SYSDATE);  
142
```

Below the code, the 'Script Output' window shows the message: "Task completed in 0,195 seconds". The 'PL/SQL procedure successfully completed.' message is displayed in the 'Output' window. The 'Dbms Output' window contains the following log output:

```
S-a actualizat reparatia.  
Informatiile despre vehicul au fost actualizate.  
Istoricul vehiculului a fost actualizat.  
-----  
Toate observatiile:  
Transferat de la depoul Titan la depoul Militari, adaugata in data de: 11.01.2024  
Reparatie majora, adaugata in data de: 10.01.2025  
-----  
S-au folosit urmatoarele componente:  
1 x Parbriz Tramvai furnizate de Rader-Vogel in lotul 2/10.09.2023  
2 x Tampoane tramvai furnizate de Alstom in lotul 13/09.01.2025  
2 x Tampoane tramvai furnizate de TisTram in lotul 7/10.01.2023  
-----
```