

PROIECT SGBD

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

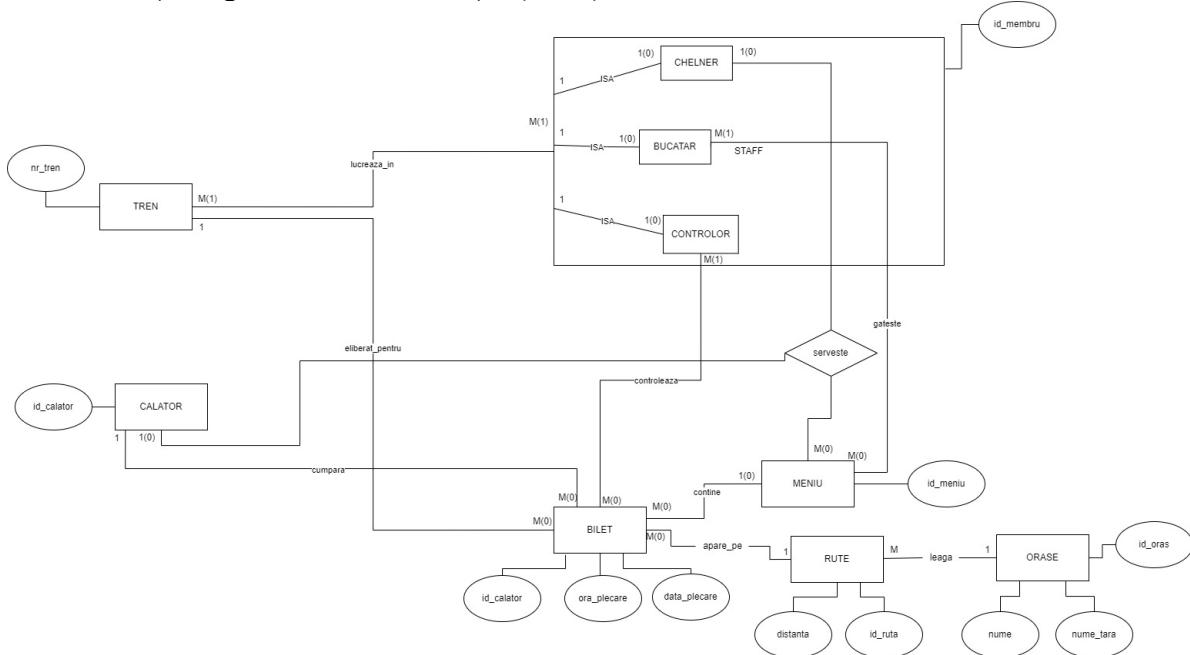
Compania "InterRails" oferă servicii de transport feroviar cetățenilor Uniunii Europene. Această companie deține un număr de trenuri care circulă între diferite orașe europene pe rutele deja existente între acestea.

Pentru fiecare tren al companiei se cunoaște numărul său, anul fabricației, numele firmei producătoare, data achiziționării și data ultimei revizii.

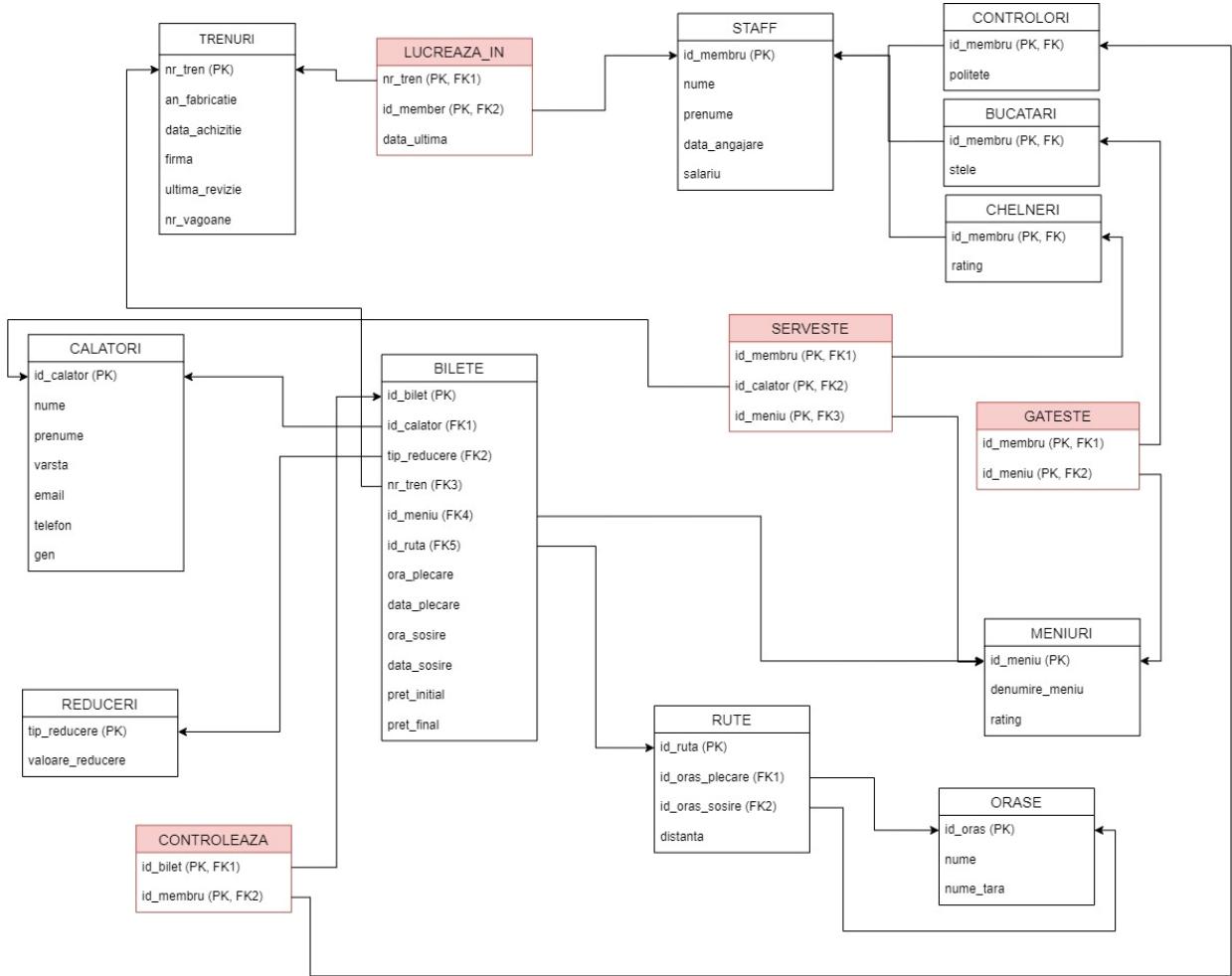
Călătorii se înregistrează cu datele personale pe platforma online a companiei, de unde achiziționează bilete. Călătorii pot dispune de diferite reduceri pentru biletele de tren. Un bilet include numele călătorului, numărul trenului, data și ora plecării, orașul de plecare, data și ora sosirii, orașul sosirii și prețul biletului respectiv (eventual și reducerea împreună cu prețul final). În plus, biletele pot include și opțiunea de meniu la vagonul de restaurant al trenului (va fi ales la cumpărarea biletului).

Aceasta companie are personal în tren (controlori, chelneri, bucătari), care asigură toate condițiile necesare unei călătorii cât mai plăcute pentru pasageri. Controlorii verifică biletele, iar chelnerii au rolul de a servi călătorilor meniurile pe care aceștia le aleg în timpul călătoriei și care sunt preparate cu încusință de către bucătarii aleși ai companiei.

2. Realizați diagrama entitate-relație (ERD).



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



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

```

create table CALATORI (
    id_calator number(8),
    nume varchar2(50) not null,
    prenume varchar2(50) not null,
    varsta number(2) not null,
    email1 varchar2(50) not null,
    email2 varchar2(50),
)

```

```

telefon varchar2(50) not null unique,
gen char(1),
constraint CALATORI_PK primary key (id_calator),
constraint gen_calatori_CK check(gen = 'M' or gen = 'F' or gen is null)
);

```

```

create table CALATORI (
    id_calator number(8),
    nume varchar2(50) not null,
    prenume varchar2(50) not null,
    varsta number(2) not null,
    email1 varchar2(50) not null,
    email2 varchar2(50),
    telefon varchar2(50) not null unique,
    gen char(1),
    constraint CALATORI_PK primary key (id_calator),
    constraint gen_calatori_CK check(gen = 'M' or gen = 'F' or gen is null)
);

create table TRENURI (
    nr_tren number(6),
    an_fabricatie number(4) default 2010,
    data_achizitie date default sysdate,
    firma varchar2(30) not null,
    ultima_revizie date,
    nr_vagoane number(2) default 1,
    constraint TRENURI_PK primary key (nr_tren),
    constraint an_fabricatie_CK check(an_fabricatie between 2010 and 2020)
);

Table CALATORI created.

```

M-am gandit sa modific structura tabelului CALATORI, insa aveam baza de date deja implementata si am ales sa o modific in felul urmator:

```
create or replace type vector_email is varray(2) of varchar(50);
```

```
/
```

```
create or replace type tabel_telefon is table of varchar(50);
```

```
/
```

```
alter table calatori add email vector_email not null;
```

```
alter table calatori add numere_telefon tabel_telefon
```

```
nested table numere_telefon store as nr_telefon;
```

```
alter table calatori drop column email1;
```

```
alter table calatori drop column email2;
```

```
alter table calatori drop column telefon;
```

Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql

```
-- asa ca am ales sa fac aceasta modificar cu alter;
/
create or replace type vector_email is varray(2) of varchar(50);
/
create or replace type tabel_telefon is table of varchar(50);
/
alter table calatori add email vector_email not null;
alter table calatori add numere_telefon tabel_telefon
nested table numere_telefon store as nr_telefon;
alter table calatori drop column email;
alter table calatori drop column email2;
alter table calatori drop column telefon;

create table TRENURI (
    nr_tren number(6),
    an_fabricatie number(4) default 2010,
    data_achizitie date default sysdate,
    firma varchar2(200) not null
)
Type VECTOR_EMAIL compiled

Type TABEL_TELEFON compiled
```

Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql

```
-- asa ca am ales sa fac aceasta modificar cu alter;
/
create or replace type vector_email is varray(2) of varchar(50);
/
create or replace type tabel_telefon is table of varchar(50);
/
alter table calatori add email vector_email not null;
alter table calatori add numere_telefon tabel_telefon
nested table numere_telefon store as nr_telefon;
alter table calatori drop column emaill;
alter table calatori drop column email2;
alter table calatori drop column telefon;

create table TRENURI (
    nr_tren number(6),
    an_fabricatie number(4) default 2010,
    data_achizitie date default sysdate,
    firma varchar2(200) not null
)
Type VECTOR_EMAIL compiled

Type TABEL_TELEFON compiled
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL Worksheet titled 'Rezolvare_Cerinte.sql' containing the following SQL code:

```

/
create or replace type tabel_telefon is table of varchar(50);
/
alter table calatori add email vector_email not null;
alter table calatori add numere_telefon tabel_telefon
nested table numere_telefon store as nr_telefon;
alter table calatori drop column email;
alter table calatori drop column email2;
alter table calatori drop column telefon;

create table TRENURI (
    nr_tren number(6),
    an_fabricatie number(4) default 2010,
    data_achizitie date default sysdate,
    firma varchar2(30) not null,
    ultima_revizie date,
    nr_vagoane number(2) default 1,
    constraint TRENURI_PK primary key (nr_tren),
    constraint an_fabricatie_CK check (an_fabricatie between 2010 and 2022),
    constraint nr_vagoane_CK check (nr_vagoane between 1 and 10)
);

```

The code is being run in the 'Script Output' tab, which shows the message 'Task completed in 0.061 seconds'. Below the code, two messages are displayed in the 'Query Result' tab: 'Table CALATORI altered.' and 'Table TRENURI altered.'.

```

create table TRENURI (
    nr_tren number(6),
    an_fabricatie number(4) default 2010,
    data_achizitie date default sysdate,
    firma varchar2(30) not null,
    ultima_revizie date,
    nr_vagoane number(2) default 1,
    constraint TRENURI_PK primary key (nr_tren),
    constraint an_fabricatie_CK check (an_fabricatie between 2010 and 2022),
    constraint nr_vagoane_CK check (nr_vagoane between 1 and 10)
);

```

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL script in the Worksheet tab. The script creates three tables: CALATORI, TRENURI, and STAFF. The STAFF table is defined with columns: id_membru (number(3)), nume (varchar2(50) not null), prenume (varchar2(50) not null), data_angajare (date default sysdate), and salariu (number(5) not null). A primary key constraint, STAFF_PK, is defined on the id_membru column. The script ends with a closing parenthesis and a semicolon. Below the script, a message indicates that the Table TRENURI was created. The status bar at the bottom right shows the current line (Line 24), column (Column 3), and task status (Task completed in 0.057 seconds).

```
constraint CALATORI_PK primary key (id_calator),
constraint gen_calatori_CK check(gen = 'M' or gen = 'F' or gen is null)
);

create table TRENURI (
    nr_tren number(6),
    an_fabricatie number(4) default 2010,
    data_achizitie date default sysdate,
    firma varchar2(30) not null,
    ultima_revizie date,
    nr_vagoane number(2) default 1,
    constraint TRENURI_PK primary key (nr_tren),
    constraint an_fabricatie_CK check (an_fabricatie between 2010 and 2022),
    constraint nr_vagoane_CK check (nr_vagoane between 1 and 10)
);

create table STAFF (
    id_membru number(3),
    nume varchar2(50) not null,
    prenume varchar2(50) not null,
    data_angajare date default sysdate,
    salariu number(5) not null
);

Table TRENURI created.
```

```
create table STAFF (
    id_membru number(3),
    nume varchar2(50) not null,
    prenume varchar2(50) not null,
    data_angajare date default sysdate,
    salariu number(5) not null,
    constraint STAFF_PK primary key (id_membru)
);
```

The screenshot shows the Oracle SQL Developer interface with the 'STAFF' table being created. The code in the Worksheet pane is:

```

constraint an_fabricatie_CK check (an_fabricatie between 2010 and 2022),
constraint nr_yagoane_CK check (nr_yagoane between 1 and 10)

);

create table STAFF (
    id_membru number(3),
    nume varchar2(50) not null,
    prenume varchar2(50) not null,
    data_angajare date default sysdate,
    salariu number(5) not null,
    constraint STAFF_PK primary key (id_membru)
);

create table CONTROLORI (
    id_membru number(3),
    politete number(1) default 1,
    constraint politete_CK check (politete between 1 and 5),
    constraint CONTROLORI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

```

The Script Output pane shows the message: "Table STAFF created."

create table CONTROLORI (

```

    id_membru number(3),
    politete number(1) default 1,
    constraint politete_CK check (politete between 1 and 5),
    constraint CONTROLORI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);
```

The screenshot shows the Oracle SQL Developer interface with the 'CONTROLORI' table being created. The code in the Worksheet pane is:

```

);

create table STAFF (
    id_membru number(3),
    nume varchar2(50) not null,
    prenume varchar2(50) not null,
    data_angajare date default sysdate,
    salariu number(5) not null,
    constraint STAFF_PK primary key (id_membru)
);

create table CONTROLORI (
    id_membru number(3),
    politete number(1) default 1,
    constraint politete_CK check (politete between 1 and 5),
    constraint CONTROLORI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

create table BUCATARI (
    id_membru number(3),
    stele number(1) default 1,
    constraint BUCATARI_CK check (stele between 1 and 5)
);

```

The Script Output pane shows the message: "Table CONTROLORI created."

```

create table BUCATARI (
    id_membru number(3),
    stele number(1) default 1,
    constraint stele_CK check (stele between 1 and 5),
    constraint BUCATARI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

```

The screenshot shows the Oracle SQL Developer interface. The code editor contains the following SQL statements:

```

create table CONTROLORI (
    id_membru number(3),
    politete number(1) default 1,
    constraint politete_CK check (politete between 1 and 5),
    constraint CONTROLORI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

create table BUCATARI (
    id_membru number(3),
    stele number(1) default 1,
    constraint stele_CK check (stele between 1 and 5),
    constraint BUCATARI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

create table CHELNERI (
    id_membru number(3),
    rating number(1) default 1,
    constraint rating_cheler_CK check (rating between 1 and 5),
    constraint CHELNERI_PK primary key (id_membru)
);

```

The status bar at the bottom right indicates "Task completed in 0.045 seconds".

```

create table CHELNERI (
    id_membru number(3),
    rating number(1) default 1,
    constraint rating_cheler_CK check (rating between 1 and 5),
    constraint CHELNERI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

```

The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code editor contains the following SQL script:

```

create table BUCATARI (
    id_membru number(3),
    stele number(1) default 1,
    constraint stele_CK check (stele between 1 and 5),
    constraint BUCATARI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

create table CHELNERI (
    id_membru number(3),
    rating number(1) default 1,
    constraint rating_cheler_CK check (rating between 1 and 5),
    constraint CHELNERI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

create table MENIURI (
    id_menui number,
    denumire_menui varchar2(50),
    rating number(1) default 1,
    constraint MENIURI_PK primary key (id_menui),
    constraint rating_menui_check check (rating between 1 and 5)
);

```

The status bar at the bottom indicates "Task completed in 0.05 seconds".

```

create table MENIURI (
    id_menui number,
    denumire_menui varchar2(50),
    rating number(1) default 1,
    constraint MENIURI_PK primary key (id_menui),
    constraint rating_menui_check check (rating between 1 and 5)
);

```

The screenshot shows the Oracle SQL Developer interface with the 'Query Builder' tab selected. The code editor contains the following SQL script:

```

);
;

create table CHELNERI (
    id_membru number(3),
    rating number(1) default 1,
    constraint rating_cheler_CK check (rating between 1 and 5),
    constraint CHELNERI_PK primary key (id_membru),
    foreign key (id_membru) references staff(id_membru) on delete cascade
);

create table MENIURI (
    id_menui number,
    denumire_menui varchar2(50),
    rating number(1) default 1,
    constraint MENIURI_PK primary key (id_menui),
    constraint rating_menui_check check (rating between 1 and 5)
);

create table REDUCERI (
    tip_reducere varchar(20),
    valoare_reducere number(3, 2),
    constraint valoare_reducere_pk primary key (valoare_reducere)
);

```

The status bar at the bottom indicates "Task completed in 0.049 seconds".

```

create table REDUCERI (
    tip_reducere varchar(20),
    valoare_reducere number(3, 2),
    constraint REDUCERI_PK primary key (tip_reducere),
    constraint valoare_reducere_CK check(valoare_reducere between 0 and 1)
);

```

```

create table MENIURI (
    id_meniu number,
    denumire_meniu varchar2(50),
    rating number(1) default 1,
    constraint MENIURI_PK primary key (id_meniu),
    constraint rating_meniu_check check (rating between 1 and 5)
);

create table REDUCERI (
    tip_reducere varchar(20),
    valoare_reducere number(3, 2),
    constraint REDUCERI_PK primary key (tip_reducere),
    constraint valoare_reducere_CK check(valoare_reducere between 0 and 1)
);

create table ORASE (
    id_oras varchar(10),
    nume varchar2(40) not null,
    nume_tara varchar(40) not null,
    constraint orase_pk primary key (id_oras)
);

Table REDUCERI created.

```

```

create table ORASE (
    id_oras varchar(10),
    nume varchar2(40) not null,
    nume_tara varchar(40) not null,
    constraint orase_pk primary key (id_oras)
);

```

```

create table REDUCERI (
    tip_reducere varchar(20),
    valoare_reducere number(3, 2),
    constraint REDUCERI_PK primary key (tip_reducere),
    constraint valoare_reducere_CK check(valoare_reducere between 0 and 1)
);

create table ORASE (
    id_oras varchar(10),
    nume varchar2(40) not null,
    nume_tara varchar(40) not null,
    constraint orase_pk primary key (id_oras)
);

create table RUTE (
    id_ruta number,
    id_oras_plecare varchar(20),
    id_oras_sosire varchar(20),
    distanta number,
    constraint ruta_pk primary key (id_ruta),
    constraint oras_plecare_fk foreign key (id_oras_plecare) references orase(id_oras),
    constraint oras_sosire_fk foreign key (id_oras_sosire) references orase(id_oras),
    constraint chk_orase_diferite check(id_oras_plecare <> id_oras_sosire)
);

```

Table ORASE created.

```

create table RUTE (
    id_ruta number,
    id_oras_plecare varchar(20),
    id_oras_sosire varchar(20),
    distanta number,
    constraint ruta_pk primary key (id_ruta),
    constraint oras_plecare_fk foreign key (id_oras_plecare) references orase(id_oras),
    constraint oras_sosire_fk foreign key (id_oras_sosire) references orase(id_oras),
    constraint chk_orase_diferite check(id_oras_plecare <> id_oras_sosire)
);

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a 'Worksheet' tab with the following SQL code:

```

create table RULE (
    id_ruta number,
    id_oras_plecare varchar(20),
    id_oras_sosire varchar(20),
    distanta number,
    constraint rule_pk primary key (id_ruta),
    constraint oras_plecare_fk foreign key (id_oras_plecare) references orase(id_oras),
    constraint oras_sosire_fk foreign key (id_oras_sosire) references orase(id_oras),
    constraint chk_orase_diferite check(id_oras_plecare <> id_oras_sosire)
);

create table BILETE (
    id_bilet number(8),
    id_calator number(8),
    tip_reducere varchar2(20),
    nr_tren number(6) not null,
    id_menui number,
    id_ruta number not null,
    ora_plecare char(5),
    data_plecare date
);

```

Below the code, the message "Table RULE created." is displayed. At the bottom of the window, the status bar shows "Task completed in 0.058 seconds". The bottom right corner of the screen shows the Windows taskbar with icons for Start, Search, Task View, File Explorer, Edge, Google Chrome, File Manager, and Task Manager.

```

create table BILETE (
    id_bilet number(8),
    id_calator number(8),
    tip_reducere varchar2(20),
    nr_tren number(6) not null,
    id_menui number,
    id_ruta number not null,
    ora_plecare char(5),
    data_plecare date,
    ora_sosire char(5) not null,
    data_sosire date not null,
    pret_initial number(5, 2) not null,
    pret_final number(5, 2) not null,
    constraint BILETE_PK primary key (id_bilet),
    foreign key (id_calator) references CALATORI(id_calator) on delete cascade,

```

foreign key (tip_reducere) references REDUCERI(tip_reducere) on delete cascade,
 foreign key (nr_tren) references TRENURI(nr_tren) on delete cascade,
 foreign key (id_menuiu) references MENIURI(id_menuiu) on delete set null,
 foreign key (id_ruta) references RUTE(id_ruta) on delete cascade
);

```

create table BILETE (
    id_bilet number(8),
    id_calator number(8),
    tip_reducere varchar2(20),
    nr_tren number(6) not null,
    id_menuiu number,
    id_ruta number not null,
    ora_plecare char(5),
    data_plecare date,
    ora_sosire char(5) not null,
    data_sosire date not null,
    pret_initial number(5, 2) not null,
    pret_final number(5, 2) not null,
    constraint BILETE_PK primary key (id_bilet),
    foreign key (id_calator) references CALATORI(id_calator) on delete cascade,
    foreign key (tip_reducere) references REDUCERI(tip_reducere) on delete cascade,
    foreign key (nr_tren) references TRENURI(nr_tren) on delete cascade,
    foreign key (id_menuiu) references MENIURI(id_menuiu) on delete set null,
    foreign key (id_ruta) references RUTE(id_ruta) on delete cascade
);

```

create table LUCREAZA_IN (
 nr_tren number(6),
 id_membru number(3),
 data_ultima date default sysdate,
 constraint LUCREAZA_IN_PK primary key (nr_tren, id_membru),
 foreign key (id_membru) references STAFF(id_membru) on delete cascade,
 foreign key (nr_tren) references TRENURI(nr_tren) on delete cascade
);

```

foreign key (id_calator) references CALATORI(id_calator) on delete cascade,
foreign key (tip_reducere) references REDUCERI(tip_reducere) on delete cascade,
foreign key (nr_tren) references TRENURI(nr_tren) on delete cascade,
foreign key (id_menuiu) references MENIURI(id_menuiu) on delete set null,
foreign key (id_ruta) references RUTE(id_ruta) on delete cascade
);

create table LUCREAZA_IN (
    nr_tren number(6),
    id_membru number(3),
    data_ultima date default sysdate,
    constraint LUCREAZA_IN_PK primary key (nr_tren, id_membru),
    foreign key (id_membru) references STAFF(id_membru) on delete cascade,
    foreign key (nr_tren) references TRENURI(nr_tren) on delete cascade
);

create table SERVESTE (
    id_membru number(3),
    id_calator number(8),
    id_menuiu number,
    constraint SERVESTE_PK primary key (id_membru, id_calator, id_menuiu),
);

```

Table LUCREAZA_IN created.

```

create table SERVESTE (
    id_membru number(3),
    id_calator number(8),
    id_menuiu number,
    constraint SERVESTE_PK primary key (id_membru, id_calator, id_menuiu),
    foreign key (id_membru) references CHELNERI(id_membru) on delete cascade,
    foreign key (id_calator) references CALATORI(id_calator) on delete cascade,
    foreign key (id_menuiu) references MENIURI(id_menuiu) on delete cascade
);

```

Oracle SQL Developer : SGBD_Project

Worksheet: Query Builder

```

foreign key (id_membru) references STAFF(id_membru) on delete cascade,
foreign key (nr_tren) references TRENURI(nr_tren) on delete cascade
);

create table SERVESTE (
    id_membru number(3),
    id_calator number(8),
    id_menui number,
    constraint SERVESTE_PK primary key (id_membru, id_calator, id_menui),
    foreign key (id_membru) references CHELNERI(id_membru) on delete cascade,
    foreign key (id_calator) references CALATORI(id_calator) on delete cascade,
    foreign key (id_menui) references MENIURI(id_menui) on delete cascade
);

create table GATESTE (
    id_membru number(3),
    id_menui varchar2(50),
    constraint GATESTE_PK primary key (id_membru, id_menui),
    foreign key (id_membru) references bucatari(id_membru) on delete cascade,
    foreign key (id_menui) references meniuri(id_menui) on delete cascade
);

```

Script Output: Task completed in 0.045 seconds

Table SERVESTE created.

create table GATESTE (

```

id_membru number(3),
id_menui varchar2(50),
constraint GATESTE_PK primary key (id_membru, id_menui),
foreign key (id_membru) references bucatari(id_membru) on delete cascade,
foreign key (id_menui) references meniuri(id_menui) on delete cascade

```

);

Oracle SQL Developer : SGBD_Project

Worksheet: Query Builder

```

foreign key (id_membru) references CHELNERI(id_membru) on delete cascade,
foreign key (id_calator) references CALATORI(id_calator) on delete cascade,
foreign key (id_menui) references MENIURI(id_menui) on delete cascade
);

create table GATESTE (
    id_membru number(3),
    id_menui number,
    constraint GATESTE_PK primary key (id_membru, id_menui),
    foreign key (id_membru) references bucatari(id_membru) on delete cascade,
    foreign key (id_menui) references meniuri(id_menui) on delete cascade
);

create table CONTROLEAZA(
    id_bilet number(8),
    id_membru number(3),
    constraint controlaza_pk primary key (id_bilet, id_membru),
    foreign key (id_bilet) references bilete(id_bilet) on delete cascade,
    foreign key (id_membru) references controlori(id_membru) on delete cascade
);

create sequence seq_inserare nocycle;

```

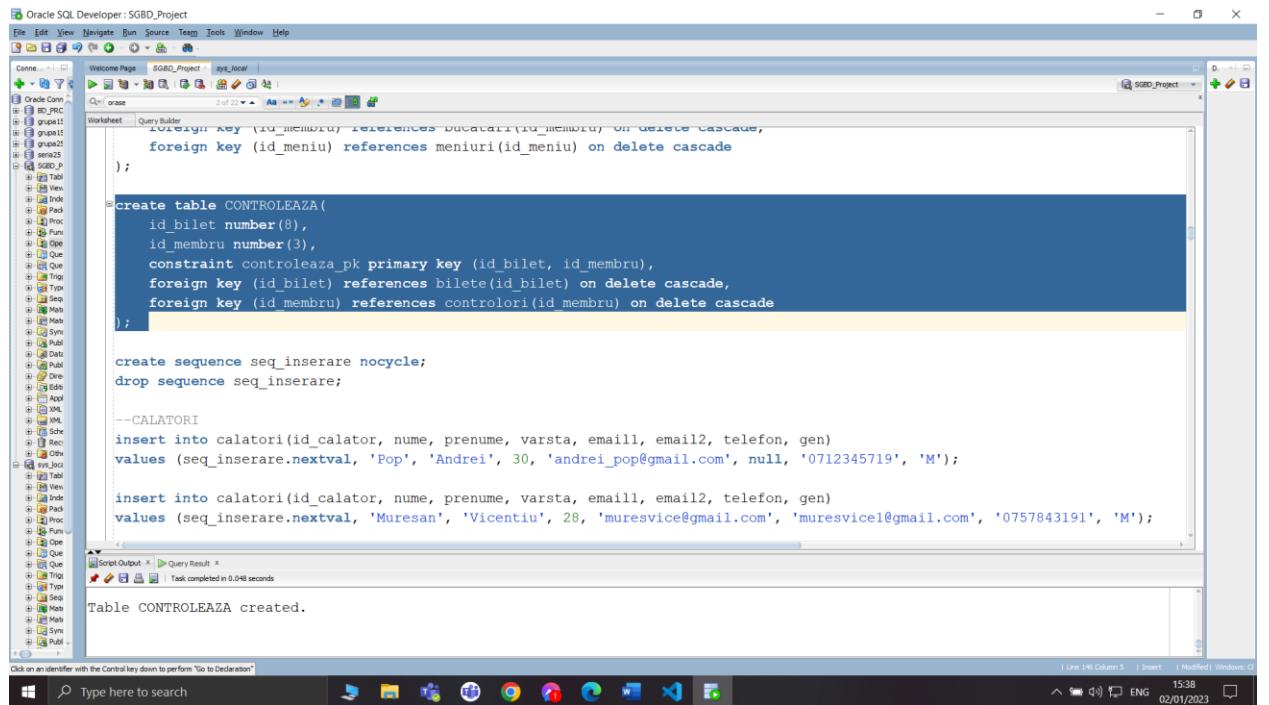
Script Output: Task completed in 0.045 seconds

Table GATESTE created.

```

create table CONTROLEAZA(
    id_bilet number(8),
    id_membru number(3),
    constraint controleaza_pk primary key (id_bilet, id_membru),
    foreign key (id_bilet) references bilete(id_bilet) on delete cascade,
    foreign key (id_membru) references controlori(id_membru) on delete cascade
);

```



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

- Project:** SGBD_Project
- Worksheet:** Query Builder
- SQL Statement:**

```

CREATE TABLE CONTROLEAZA(
    id_bilet NUMBER(8),
    id_membru NUMBER(3),
    CONSTRAINT controleaza_pk PRIMARY KEY (id_bilet, id_membru),
    FOREIGN KEY (id_bilet) REFERENCES bilete(id_bilet) ON DELETE CASCADE,
    FOREIGN KEY (id_membru) REFERENCES controlori(id_membru) ON DELETE CASCADE
);

CREATE SEQUENCE seq_inserare NOCYCLE;
DROP SEQUENCE seq_inserare;

--CALATORI
INSERT INTO calatori(id_calator, nume, prenume, varsta, email1, email2, telefon, gen)
VALUES (seq_inserare.nextval, 'Pop', 'Andrei', 30, 'andrei_pop@gmail.com', null, '0712345719', 'M');

INSERT INTO calatori(id_calator, nume, prenume, varsta, email1, email2, telefon, gen)
VALUES (seq_inserare.nextval, 'Muresan', 'Vicentiu', 28, 'muresvice@gmail.com', 'muresvice1@gmail.com', '0757843191', 'M');

```
- Script Output:** Task completed in 0.048 seconds
- Message:** Table CONTROLEAZA created.
- System Status:** Line 16 Column 5 Insert 1 Modified 1 Windows

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

--CALATORI

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (19, 'Pop', 'Andrei', 30, vector_email('andrei_pop@gmail.com'),
tabel_telefon('0712345719'), 'M');
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (20, 'Muresan', 'Vicentiu', 28, vector_email('muresvice@gmail.com',
'muresvice1@gmail.com'), tabel_telefon('0757843191'), 'M');
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (21, 'Constantinescu', 'Catalin', 18, vector_email('cataconstantin@gmail.com'),
tabel_telefon('0712345718'), null);
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (22, 'Pindaru', 'Adelina', 40, vector_email('pindaruAdelina@gmail.com',
'adelinapin@gmail.com'), tabel_telefon('0712345710'), 'F');
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (23, 'Pavel', 'Doina', 45, vector_email('doina_pavel@gmail.com'),
tabel_telefon('0749135476'), 'F');
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (24, 'Dobrica', 'Casiana-Elena', 25, vector_email('casyelena@gmail.com',
'elena.casiana-dobrica@gmail.com'), tabel_telefon('0772149658'), null);
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon)
```

```
values(25,      'Savoiu',      'Mincu',      30,      vector_email('savoumin@yahoo.com'),
tabel_telefon('0716495137'));
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values(26,      'Sancu',      'Monica',      23,      vector_email('monicasan@outlook.com'),
tabel_telefon('0716496257'), 'F');
```

```
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values(27, 'Samburel', 'Alexandra-Mirela', 32, vector_email('alexymirela@gmail.com'),
tabel_telefon('0756713498'), 'F');
```

```
--5.
--CALATORI
insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (19, 'Pop', 'Andrei', 30, vector_email('andrei_pop@gmail.com'), tabel_telefon('0712345719'), 'M');

insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (20, 'Muresan', 'Vicentiu', 28, vector_email('mureservice@gmail.com', 'muresvicel@gmail.com'), tabel_telefon('0757843191'), 'M');

insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (21, 'Constantinescu', 'Catalin', 18, vector_email('cataconstantin@gmail.com'), tabel_telefon('0712345718'), null);

insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (22, 'Pindaru', 'Adelina', 40, vector_email('pindaruAdelina@gmail.com', 'adelinapin@gmail.com'), tabel_telefon('0712345710'), 'F');

insert into calatori(id_calator, nume, prenume, varsta, email, numere_telefon, gen)
values (23, 'Pavel', 'Doina', 45, vector_email('doina_pavel@gmail.com'), tabel_telefon('0749135476'), 'F');
```

--TRENURI

```
insert into trenuri(nr_tren, an_fabricatie, firma)
values(28, 2011, 'Rails2GO');
```

```
insert into trenuri
```

```
values(29, 2019, to_date('15-08-2020', 'DD-MM-YYYY'), 'TrainsForever', null, 8);
```

insert into trenuri

```
values(30, 2015, to_date('16-06-2016', 'DD-MM-YYYY'), 'TrainsForever', to_date('27-04-2018', 'DD-MM-YYYY'), 4);
```

insert into trenuri

```
values(31, 2017, to_date('15-08-2017', 'DD-MM-YYYY'), 'Rails2Go', to_date('19-05-2021', 'DD-MM-YYYY'), 2);
```

insert into trenuri

```
values(32, 2010, to_date('20-09-2019', 'DD-MM-YYYY'), 'TrailLover', null, 5);
```

insert into trenuri

```
values(75, 2017, sysdate - 30, 'TrailLover', null, 3);
```

```
commit;

--TRENURI
insert into trenuri(nr_tren, an_fabricatie, firma)
values(28, 2011, 'Rails2GO');

insert into trenuri
values(29, 2019, to_date('15-08-2020', 'DD-MM-YYYY'), 'TrainsForever', null, 8);

insert into trenuri
values(30, 2015, to_date('16-06-2016', 'DD-MM-YYYY'), 'TrainsForever', to_date('27-04-2018', 'DD-MM-YYYY'), 4);

insert into trenuri
values(31, 2017, to_date('15-08-2017', 'DD-MM-YYYY'), 'Rails2Go', to_date('19-05-2021', 'DD-MM-YYYY'), 2);

insert into trenuri
values(32, 2010, to_date('20-09-2019', 'DD-MM-YYYY'), 'TrailLover', null, 5);

1 row inserted.

1 row inserted.
```

--STAFF + BUCATARI

```
insert into staff(id_membru, nume, prenume, data_angajare, salariu)
```

```
values(33, 'Dumitru', 'Adela', to_date('15-02-2013', 'DD-MM-YYYY'), 5000);
```

```
insert into bucatari(id_membru, stele)  
values(33, 4);
```

```
insert into staff(id_membru, nume, prenume, salariu)  
values(34, 'Vlasceanu', 'Diana', 7000);
```

```
insert into bucatari(id_membru)  
values(34);
```

```
insert into staff(id_membru, nume, prenume, salariu)  
values(35, 'Dinu', 'Lucian', 6000);
```

```
insert into bucatari(id_membru, stele)  
values(35, 5);
```

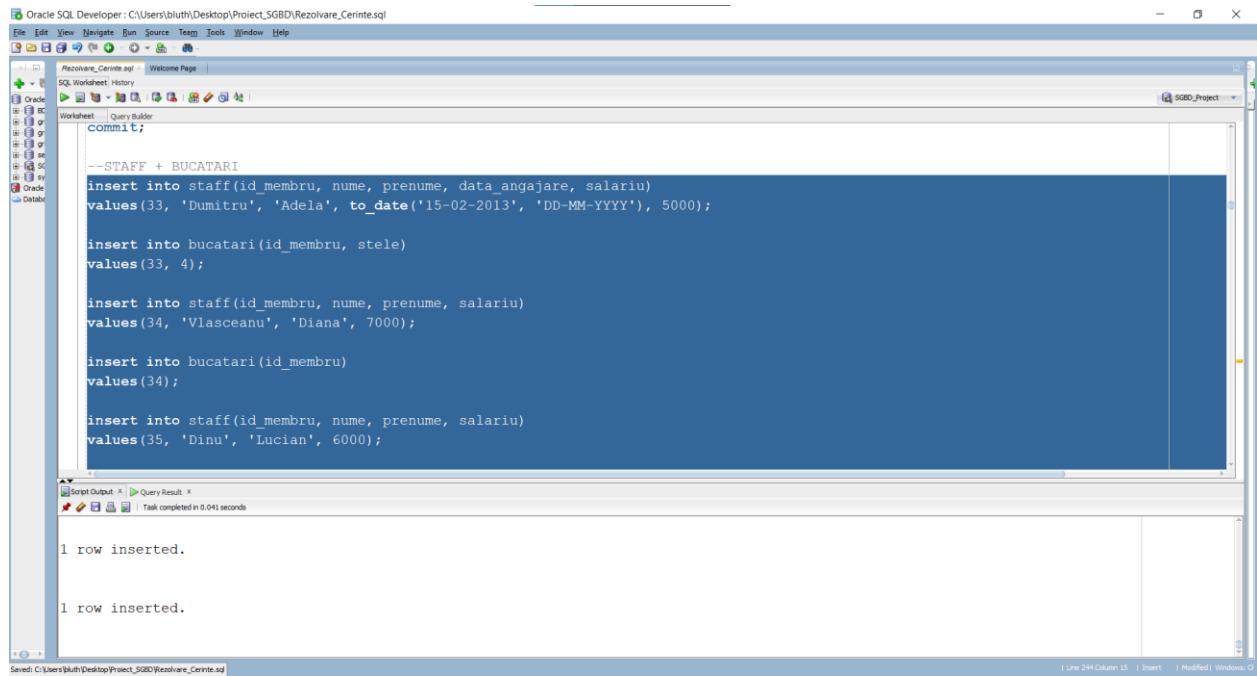
```
insert into staff(id_membru, nume, prenume, salariu)  
values(36, 'Ifrim', 'Andrei', 10000);
```

```
insert into bucatari(id_membru, stele)  
values(36, 2);
```

```
insert into staff(id_membru, nume, prenume, salariu)  
values(37, 'Stanciu', 'Horia', 12000);
```

```
insert into bucatari(id_membru, stele)
```

values(37, 5);



```
--STAFF + BUCATARISave: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql
insert into staff(id_membru, nume, prenume, data_angajare, salariu)
values(33, 'dumitru', 'Adela', to_date('15-02-2013', 'DD-MM-YYYY'), 5000);

insert into bucatari(id_membru, stele)
values(33, 4);

insert into staff(id_membru, nume, prenume, salariu)
values(34, 'Vlasceanu', 'Diana', 7000);

insert into bucatari(id_membru)
values(34);

insert into staff(id_membru, nume, prenume, salariu)
values(35, 'Dinu', 'Lucian', 6000);

1 row inserted.

1 row inserted.
```

--STAFF + CONTROLORI

```
insert into staff(id_membru, nume, prenume, salariu)
values(38, 'Sava', 'Alexandru', 5500);
```

```
insert into controlorii(id_membru, politete)
values(38, 5);
```

```
insert into staff(id_membru, nume, prenume, salariu)
values(39, 'Manole', 'Iulian-George', 6500);
```

```
insert into controlorii(id_membru)
values(39);
```

```
insert into staff(id_membru, nume, prenume, salariu)
```

```
values(40, 'Popescu', 'Madalina-Andreea', 7800);
```

```
insert into controlorii(id_membru, politete)
values(40, 4);
```

```
insert into staff(id_membru, nume, prenume, salariu)
values(41, 'Nita', 'Loredana Monica', 4500);
```

```
insert into controlorii(id_membru)
values(41);
```

```
insert into staff(id_membru, nume, prenume, salariu)
values(42, 'Tomescu', 'Sebastian Emil', 9000);
```

```
insert into controlorii(id_membru, politete)
values(42, 3);
```

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

- Title Bar:** Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql
- Toolbar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Left Sidebar:** Shows the connection to the 'SGBD' database.
- Worksheet:** Contains the SQL script code.
- Script Output:** Shows the results of the executed statements.
- Status Bar:** Shows the file path 'C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql' and the status 'Line 277 Column 15 | Insert | Modified | Windows'.

```
--STAFF + CONTROLORI
insert into staff(id_membru, nume, prenume, salariu)
values(38, 'Sava', 'Alexandru', 5500);

insert into controlorii(id_membru, politete)
values(38, 5);

insert into staff(id_membru, nume, prenume, salariu)
values(39, 'Manole', 'Iulian-George', 6500);

insert into controlorii(id_membru)
values(39);

insert into staff(id_membru, nume, prenume, salariu)
values(40, 'Popescu', 'Madalina-Andreea', 7800);

commit;
```

Script Output: Task completed in 0.041 seconds

1 row inserted.

1 row inserted.

--STAFF + CHELNERI

```
insert into staff(id_membru, nume, prenume, salariu)
values(43, 'Pavel', 'Narcisa', 4000);
```

```
insert into chelneri(id_membru, rating)
values(43, 2);
```

```
insert into staff(id_membru, nume, prenume, salariu)
values(44, 'Olteanu', 'Stefania-Maria', 6500);
```

```
insert into chelneri(id_membru, rating)
values(44, 4);
```

```
insert into staff(id_membru, nume, prenume, salariu)
values(45, 'Ciocu', 'Bogdan Constantin', 7500);
```

```
insert into chelneri(id_membru, rating)
values(45, 5);
```

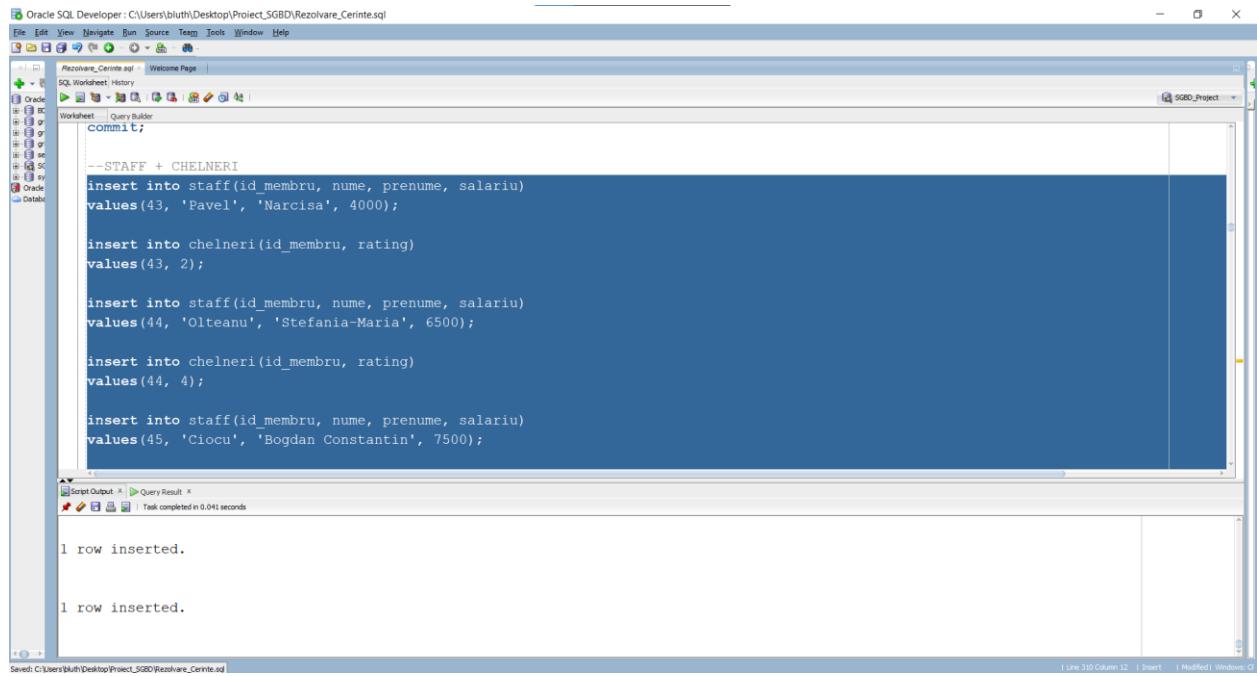
```
insert into staff(id_membru, nume, prenume, salariu)
values(46, 'Craciun', 'Cristina', 8000);
```

```
insert into chelneri(id_membru, rating)
values(46, 5);
```

```
insert into staff(id_membru, nume, prenume, salariu)
```

```
values(47, 'Pascu', 'Mirel Petre', 3500);
```

```
insert into chelneri(id_membru)
values(47);
```



```
-- Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql
File Edit View Navigate Run Source Tools Window Help
Rezolvare_Cerinte.sql - Welcome Page | Oracle Database
SQL Worksheet History
Worksheet: Query Builder
COMMIT;
--STAFF + CHELNERI
insert into staff(id_membru, nume, prenume, salariu)
values(43, 'Pavel', 'Narcisa', 4000);

insert into chelneri(id_membru, rating)
values(43, 2);

insert into staff(id_membru, nume, prenume, salariu)
values(44, 'Olteanu', 'Stefania-Maria', 6500);

insert into chelneri(id_membru, rating)
values(44, 4);

insert into staff(id_membru, nume, prenume, salariu)
values(45, 'Ciocu', 'Bogdan Constantin', 7500);

Script Output: X | Query Result: X | Task completed in 0.041 seconds
1 row inserted.

1 row inserted.

Saved: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql | Line 310 Column 12 | Insert | Modified | Windows: O
```

--MENIURI

```
insert into meniuri(id_menu, denumire_menu, rating)
values(48, 'Pulpe de pui la cuptor cu cartofi si salata', 4);
```

```
insert into meniuri(id_menu, denumire_menu, rating)
values(49, 'Piure de cartofi cu friptura de porc', 5);
```

```
insert into meniuri(id_menu, denumire_menu, rating)
values(50, 'Ciorba de fasole in paine', 5);
```

```
insert into meniuri(id_meniu, denumire_meniu, rating)
values(51, 'Paste carbonara', 2);
```

```
insert into meniuri(id_meniu, denumire_meniu, rating)
values(52, 'Paella', 2);
```

```
insert into meniuri(id_meniu, denumire_meniu, rating)
values(53, 'Cartofi prajiti cu piept de pui la gratar', 3);
```

```
Oracle SQL Developer : C:\Users\bluth\Desktop\Proiect_SGBD\Rezolvare_Cerinte.sql
File Edit View Navigate Run Source Team Tools Window Help
Rezolvare_Cerinte.sql | Welcome Page | SQL Worksheet History
+ Oracle
+ [ ] n
+ [ ] o
+ [ ] g
+ [ ] b
+ [ ] s
+ [ ] w
+ [ ] Oracle
+ [ ] Database
Worksheet | Query Builder
commit;
:
:
--MENIURI
insert into meniuri(id_meniu, denumire_meniu, rating)
values(48, 'Pulpe de pui la cuptor cu cartofii si salata', 4);

insert into meniuri(id_meniu, denumire_meniu, rating)
values(49, 'Piure de cartofii cu friptura de porc', 5);

insert into meniuri(id_meniu, denumire_meniu, rating)
values(50, 'Ciorba de fasole in paine', 5);

insert into meniuri(id_meniu, denumire_meniu, rating)
values(51, 'Paste carbonara', 2);

insert into meniuri(id_meniu, denumire_meniu, rating)
values(52, 'Paella', 2);
1 row inserted.

1 row inserted.
```

--REDUCERI

```
insert into reduceri(tip_reducere, valoare_reducere)
values('Student 50%', 0.5);
```

```
insert into reduceri(tip_reducere, valoare_reducere)
values('Elev 50%', 0.5);
```

```
insert into reduceri(tip_reducere, valoare_reducere)
```

```
values('Donator sange 25%', 0.25);
```

```
insert into reduceri(tip_reducere, valoare_reducere)
values('Veteran Razboi 75%', 0.75);
```

```
insert into reduceri(tip_reducere, valoare_reducere)
values('Nevoi speciale 50%', 0.5);
```

```
--REDUCERI
insert into reduceri(tip_reducere, valoare_reducere)
values('Student 50%', 0.5);

insert into reduceri(tip_reducere, valoare_reducere)
values('Elev 50%', 0.5);

insert into reduceri(tip_reducere, valoare_reducere)
values('Donator sange 25%', 0.25);

insert into reduceri(tip_reducere, valoare_reducere)
values('Veteran Razboi 75%', 0.75);

insert into reduceri(tip_reducere, valoare_reducere)
values('Nevoi speciale 50%', 0.5);
```

Output:

```
1 row inserted.

1 row inserted.
```

--ORASE

```
insert into orase(id_oras, nume, nume_tara)
values('BuchRO', 'Bucuresti', 'Romania');
```

```
insert into orase(id_oras, nume, nume_tara)
values('RomeIT', 'Roma', 'Italia');
```

```
insert into orase(id_oras, nume, nume_tara)
values('MadridSP', 'Madrid', 'Spania');
```

```
insert into orase(id_oras, nume, nume_tara)
values('ParisFR', 'Paris', 'Franta');
```

```
insert into orase(id_oras, nume, nume_tara)
values('BerlinD', 'Berlin', 'Germania');
```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays a tree view of the database schema, including Oracle Catalog, Schemas, and various tables like orase, route, and tara. The central workspace contains a query builder window with the following SQL script:

```
--ORASE
insert into orase(id_oras, nume, nume_tara)
values('BuchRO', 'Bucuresti', 'Romania');

insert into orase(id_oras, nume, nume_tara)
values('RomeIT', 'Roma', 'Italia');

insert into orase(id_oras, nume, nume_tara)
values('MadridSP', 'Madrid', 'Spania');

insert into orase(id_oras, nume, nume_tara)
values('ParisFR', 'Paris', 'Franta');

insert into orase(id_oras, nume, nume_tara)
values('BerlinD', 'Berlin', 'Germania');
```

Below the workspace is a Script Output tab showing the results of the execution:

```
1 row inserted.

1 row inserted.
```

A status bar at the bottom indicates "Task completed in 0.065 seconds".

--RUTE

```
insert into rute values(55, 'BuchRO', 'RomeIT', 2050);
insert into rute values(56, 'BerlinD', 'RomeIT', 1600);
insert into rute values(57, 'MadridSP', 'ParisFR', 1300);
insert into rute values(58, 'ParisFR', 'MadridSP', 1300);
insert into rute values(59, 'BuchRO', 'BerlinD', 1700);
```

```

-- Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql
File Edit View Navigate Run Source Team Tools Window Help
Oracle Database Navigator SQL Worksheet History
Worksheet - Query Builder
Rezolvare_Cerinte.sql | Welcome Page |
insert into orase(id_oras, nume, nume_tara)
values('BerlinD', 'Berlin', 'Germania');

commit;

--ROUTE
insert into ruta values(55, 'BuchRO', 'RomeIT', 2050);
insert into ruta values(56, 'BerlinD', 'RomeIT', 1600);
insert into ruta values(57, 'MadridSP', 'ParisFR', 1300);
insert into ruta values(58, 'ParisFR', 'MadridSP', 1300);
insert into ruta values(59, 'BuchRO', 'BerlinD', 1700);

commit;

--BILETE
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)

Script Output | Query Result | Task completed in 0.041 seconds
1 row inserted.

1 row inserted.

```

Saved: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql

--BILETE

```

insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire,
pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)

values(67, 19, '10:50', to_date('18-06-2022', 'DD-MM-YYYY'), '10:50', to_date('18-06-
2022', 'DD-MM-YYYY') + 1, 70.50, null, 70.50, 28, 48, 55);

```

```

insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire,
pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)

```

```

values(68, 20, '06:57', to_date('18-06-2022', 'DD-MM-YYYY'), '00:26', to_date('18-06-
2022', 'DD-MM-YYYY') + 1, 100, 'Elev 50%', 50, 29, 52, 56);

```

```

insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire,
pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)

```

```

values(69, 23, '08:49', to_date('18-06-2022', 'DD-MM-YYYY'), '23:56', to_date('18-06-
2022', 'DD-MM-YYYY'), 80, 'Student 50%', 40, 30, null, 57);

```

```

insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire,
pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)

```

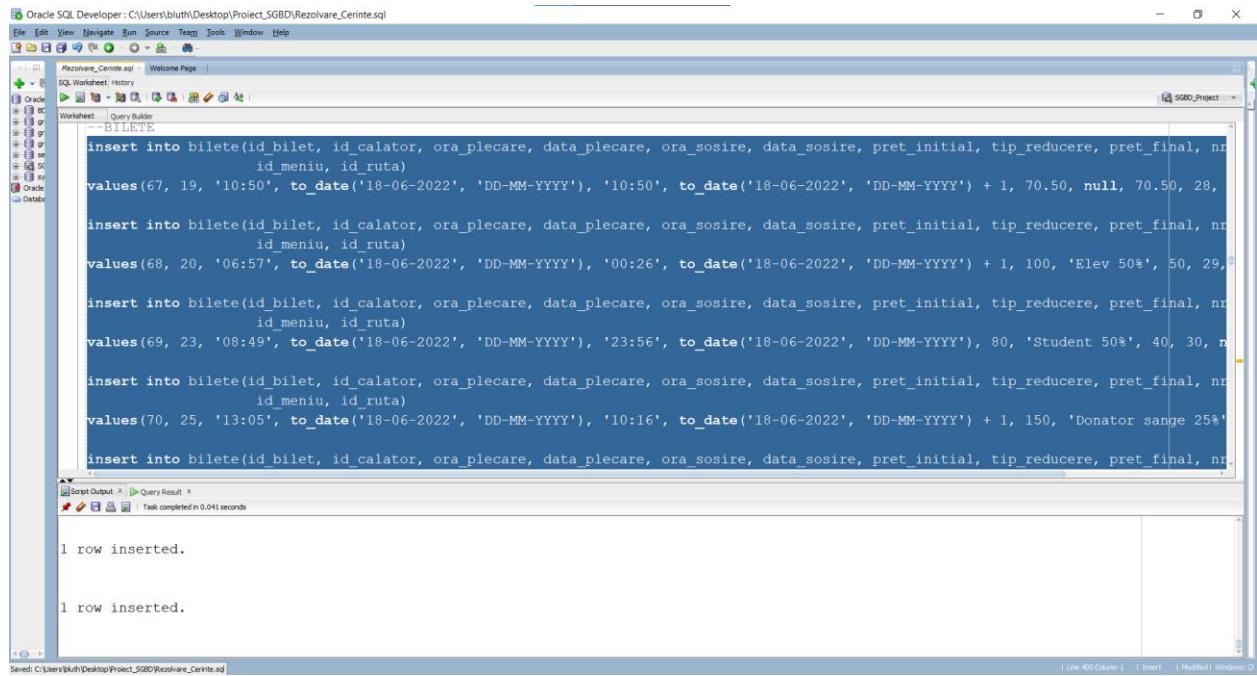
```
values(70, 25, '13:05', to_date('18-06-2022', 'DD-MM-YYYY'), '10:16', to_date('18-06-2022', 'DD-MM-YYYY') + 1, 150, 'Donator sange 25%', 112.5, 31, 52, 58);
```

```
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
```

```
values(71, 26, '05:36', to_date('12-07-2019', 'DD-MM-YYYY'), '22:41', to_date('13-07-2019', 'DD-MM-YYYY') + 1, 120, null, 120, 32, null, 59);
```

```
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
```

```
values(72, 24, '07:20', to_date('15-08-2020', 'DD-MM-YYYY'), '03:25', to_date('15-08-2020', 'DD-MM-YYYY') + 1, 110, 'Elev 50%', 55, 28, 49, 55);
```



```
Rezolvare_Cerinte.sql - Welcome Page | Oracle SQL Developer: C:\Users\blith\Desktop\Project_SGBD\Rezolvare_Cerinte.sql
File Edit View Navigate Run Source Text Tools Window Help
SQL Worksheet History
Oracle Database
-- BILETE
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
values(67, 19, '10:50', to_date('18-06-2022', 'DD-MM-YYYY'), '10:50', to_date('18-06-2022', 'DD-MM-YYYY') + 1, 70.50, null, 70.50, 28, null, null)
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
values(68, 20, '06:57', to_date('18-06-2022', 'DD-MM-YYYY'), '00:26', to_date('18-06-2022', 'DD-MM-YYYY') + 1, 100, 'Elev 50%', 50, 29, null, null)
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
values(69, 23, '08:49', to_date('18-06-2022', 'DD-MM-YYYY'), '23:56', to_date('18-06-2022', 'DD-MM-YYYY'), 80, 'Student 50%', 40, 30, null, null)
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
values(70, 25, '13:05', to_date('18-06-2022', 'DD-MM-YYYY'), '10:16', to_date('18-06-2022', 'DD-MM-YYYY') + 1, 150, 'Donator sange 25%', 112.5, 31, 52, 58)
insert into bilet(id_bilet, id_calator, ora_plecare, data_plecare, ora_sosire, data_sosire, pret_initial, tip_reducere, pret_final, nr_tren, id_meniu, id_ruta)
values(71, 26, '05:36', to_date('12-07-2019', 'DD-MM-YYYY'), '22:41', to_date('13-07-2019', 'DD-MM-YYYY') + 1, 120, null, 120, 32, null, 59)

Script Output | Query Result
1 row inserted.
1 row inserted.

Task completed in 0.041 seconds
Line 403 Column 1 | Insert | Modified | Windows |
```

--LUCREAZA_IN

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
```

```
values(28, 38, to_date('25-03-2018', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(29, 39, to_date('25-03-2018', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(30, 41, to_date('26-04-2019', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(28, 45, to_date('02-01-2023', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(32, 46, to_date('10-11-2019', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(31, 39, to_date('02-01-2023', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(30, 43, to_date('02-01-2023', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(29, 47, to_date('06-12-2021', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(31, 40, to_date('05-02-2019', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(32, 45, to_date('02-01-2023', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(28, 33, to_date('13-05-2022','DD-MM-YYYY'));
```

```
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(31, 37, to_date('13-05-2022','DD-MM-YYYY'));
```

```
insert into lucreaza_in (id_membru, nr_tren, data_ultima)
values(39, 75, to_date('04-01-2023', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in (id_membru, nr_tren, data_ultima)
values(40, 75, to_date('04-01-2023', 'DD-MM-YYYY'));
```

```
insert into lucreaza_in(id_membru, nr_tren, data_ultima)
values(35, 28, to_date('07-01-2023', 'DD-MM-YYYY'));
```

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql". The main window displays a script titled "Rezolvare_Cerinte.sql" with the following content:

```
-- LUCREAZA_IN
insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(28, 38, to_date('25-03-2018', 'DD-MM-YYYY'));

insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(29, 39, to_date('25-03-2018', 'DD-MM-YYYY'));

insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(30, 41, to_date('26-04-2019', 'DD-MM-YYYY'));

insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(28, 45, to_date('02-01-2023', 'DD-MM-YYYY'));

insert into lucreaza_in(nr_tren, id_membru, data_ultima)
values(32, 46, to_date('10-11-2019', 'DD-MM-YYYY'));

insert into lucreaza_in(nr_tren, id_membru, data_ultima)
```

Below the script, the "Script Output" tab shows the results of the execution:

```
1 row inserted.

1 row inserted.
```

The status bar at the bottom indicates "Saved: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql" and "Line 438 Column 1".

--SERVESTE

```
insert into serveste(id_membru, id_calator, id_menui)
values(45, 21, 48);

insert into serveste(id_membru, id_calator, id_menui)
values(45, 25, 50);

insert into serveste(id_membru, id_calator, id_menui)
values(47, 22, 49);

insert into serveste(id_membru, id_calator, id_menui)
values(43, 24, 52);

insert into serveste(id_membru, id_calator, id_menui)
values(45, 21, 51);

insert into serveste(id_membru, id_calator, id_menui)
values(43, 24, 53);

insert into serveste(id_membru, id_calator, id_menui)
values(47, 25, 49);

insert into serveste(id_membru, id_calator, id_menui)
values(46, 23, 48);

insert into serveste(id_membru, id_calator, id_menui)
values(47, 26, 50);

insert into serveste(id_membru, id_calator, id_menui)
values(44, 25, 50);
```

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql". The main window displays a SQL Worksheet titled "Rezolvare_Cerinte.sql" with the following code:

```
--SERVESTE
insert into serveste(id_membru, id_calator, id_menu)
values(45, 21, 48);
insert into serveste(id_membru, id_calator, id_menu)
values(45, 25, 50);
insert into serveste(id_membru, id_calator, id_menu)
values(47, 22, 49);
insert into serveste(id_membru, id_calator, id_menu)
values(43, 24, 52);
insert into serveste(id_membru, id_calator, id_menu)
values(45, 21, 51);
insert into serveste(id_membru, id_calator, id_menu)
values(43, 24, 53);
insert into serveste(id_membru, id_calator, id_menu)
values(47, 25, 49);
insert into serveste(id_membru, id_calator, id_menu)
values(46, 23, 48);
```

The output pane shows the results of the execution:

```
1 row inserted.

1 row inserted.
```

At the bottom, the status bar indicates "Saved: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql" and "Line 474 Column 32 | Insert | Modified | Windows: 0".

--GATESTE

```
insert into gateste(id_membru, id_menu)
values(36, 50);

insert into gateste(id_membru, id_menu)
values(33, 52);

insert into gateste(id_membru, id_menu)
values(34, 51);

insert into gateste(id_membru, id_menu)
values(33, 48);

insert into gateste(id_membru, id_menu)
values(35, 50);

insert into gateste(id_membru, id_menu)
values(34, 49);

insert into gateste(id_membru, id_menu)
values(34, 48);
```

```

insert into gateste(id_membru, id_menui)
values(35, 53);

insert into gateste(id_membru, id_menui)
values(36, 52);

insert into gateste(id_membru, id_menui)
values(37, 53);

insert into gateste(id_membru, id_menui)
values(36, 53);

```

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql". The main area displays an SQL script with multiple insert statements into the 'gateste' table. Below the script, the "Script Output" window shows the results of the execution: "1 row inserted." and "1 row inserted.", indicating successful insertions.

```

--GATESTE
insert into gateste(id_membru, id_menui)
values(36, 50);
insert into gateste(id_membru, id_menui)
values(33, 52);
insert into gateste(id_membru, id_menui)
values(34, 51);
insert into gateste(id_membru, id_menui)
values(33, 48);
insert into gateste(id_membru, id_menui)
values(35, 50);
insert into gateste(id_membru, id_menui)
values(34, 49);
insert into gateste(id_membru, id_menui)
values(34, 48);
insert into gateste(id_membru, id_menui)
values(35, 53);

```

--CONTROLEAZA

```

insert into controleaza(id_bilet, id_membru)
values(72, 38);

insert into controleaza(id_bilet, id_membru)
values(72, 39);

insert into controleaza(id_bilet, id_membru)
values(69, 40);

```

```

insert into controleaza(id_bilet, id_membru)
values(70, 38);

insert into controleaza(id_bilet, id_membru)
values(68, 38);

insert into controleaza(id_bilet, id_membru)
values(69, 41);

insert into controleaza(id_bilet, id_membru)
values(69, 38);

insert into controleaza(id_bilet, id_membru)
values(67, 41);

insert into controleaza(id_bilet, id_membru)
values(71, 39);

insert into controleaza(id_bilet, id_membru)
values(67, 42);

insert into controleaza(id_bilet, id_membru)
values(70, 41);

```

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

- Title Bar:** Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql
- Toolbar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Left Sidebar:** Shows Database Connections (ID_Proiect, ID_Proiect_2, grupa1, grupa2, serial25, ID_Proiect_3) and a Project named SGBD_Project.
- Central Area:** A Worksheet tab is active, displaying the SQL code for inserting data into the CONTROLEAZA table. The code is identical to the one provided in the text block above.
- Bottom Area:** Script Output tab shows the results of the execution: "1 row inserted." and "1 row inserted.".
- Status Bar:** Shows the path "Saved: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql" and line numbers "Line 530 Column 1".

MENTIUNE IMPORTANTA: Numele directorului pe care l-am folosit in exercitiile urmatoare este **FILE_DIR.**

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri de colecție studiate. Apelați subprogramul.

CERINTA: Sunt date două fisiere: 'data_phone.txt' și 'data_email.txt', ambele continând informații pe mai multe linii. Fiecare linie din fiecare fișier conține informații sub următoarea formă: <id_calator>,<informație>, unde <informație> este fie un număr de telefon, fie o adresă de email, în funcție de fișier.

Scripteți o procedură care are 2 parametri:

- operatie (varchar - 'update_phone' / 'update_email' non-case sensitive; parametru intrare)

- nr_calatori_actualizati (numarul calatorilor actualizați; id-urile se pot repeta în fișier; parametru ieșire)

Procedura va efectua următoarele operații:

a) update_phone: citeste din fișierul 'data_phone.txt' și actualizează tabelul CALATORI inserând pentru fiecare id_calator numarul respectiv de telefon în lista corespunzătoare lui din baza de date

b) update_email: citeste din fișierul 'data_email.txt' și actualizează tabelul CALATORI inserând pentru fiecare id_calator email-ul respectiv în lista corespunzătoare lui din baza de date; dacă un calator are deja 2 email-uri atunci se va întrerupe execuția procedurii în momentul respectiv.

COD:

```
create or replace procedure update_phone_email(operatie in varchar,  
nr_calatori_actualizati out nocopy number)  
is
```

```
    file_handle utl_file.file_type;  
    fline varchar(100);  
    type indexed_table is table of number index by pls_integer;  
    words extra_pack.varchar_table := extra_pack.varchar_table();  
    hash_table indexed_table;  
    words varchar_table;  
    v_email vector_email;  
    v_id_calator number;  
  
begin  
    if lower(operatie) = 'update_phone' then  
        file_handle := utl_file.fopen('FILE_DIR', 'data_phone.txt', 'r');  
        loop
```

```

begin
    utl_file.get_line(file_handle, fline);
exception
    when no_data_found then exit;
end;

words := extra_pack.sparge_string_regex(fline, '[^,]+');
v_id_calator := to_number(words(1));
if not hash_table.exists(v_id_calator) then
    hash_table(v_id_calator) := 1;
end if;

insert into table(select numere_telefon from calatori where id_calator =
v_id_calator)
values(words(2));

end loop;

nr_calatori_actualizati := hash_table.count;
utl_file.fclose(file_handle);

elsif lower(operatie) = 'update_email' then
    file_handle := utl_file=fopen('FILE_DIR', 'data_email.txt', 'r');
loop
begin
    utl_file.get_line(file_handle, fline);
exception
    when no_data_found then exit;
end;

words := extra_pack.sparge_string_regex(fline, '[^,]+');
v_id_calator := to_number(words(1));
select email into v_email from calatori where id_calator = v_id_calator;
if v_email.count = 2 then
    nr_calatori_actualizati := hash_table.count;
    raise_application_error(-20002, 'Calatorul cu id-ul ' || v_id_calator || ' are deja 2
adrese de email');
end if;

if not hash_table.exists(v_id_calator) then
    hash_table(v_id_calator) := 1;

```

```

end if;

v_email.extend;
v_email(v_email.last) := words(2);
update calatori
set email = v_email
where id_calator = v_id_calator;

end loop;

nr_calatori_actualizati := hash_table.count;
utl_file.fclose(file_handle);
else
    raise_application_error(-20001, 'Operatia "' || operatie || "' nu este cunoscuta!");
end if;
end;

```

APEL:

```

-- Am creeat un pachet cu exceptii pentru a avea la un loc exceptiile
-- definite de mine si folosite in afara subprogramelor

create or replace package exceptii is
    operatie_invalida exception;
    limita_email_depasita exception;
    pragma exception_init(operatie_invalida, -20001);
    pragma exception_init(limita_email_depasita, -20002);
end;
/
declare
    nr_return number;
begin
    update_phone_email('update_phone', nr_return);
    dbms_output.put_line('S-au actualizat ' || nr_return || ' inregistrari');
exception
    when exceptii.operatie_invalida then
        dbms_output.put_line(SQLERRM);
        dbms_output.put_line('Incercati o operatie permisa: "update_phone" / '
"update_email");
    when exceptii.limita_email_depasita then
        dbms_output.put_line(SQLERRM);
        dbms_output.put_line('S-au actualizat doar' || nr_return || ' inregistrari');

```

end;

/

The screenshot shows the Oracle SQL Developer interface with the 'Rezolvare_Cerinte.sql' script open in the 'Worksheet' tab. The code defines a procedure 'UPDATE_PHONE_EMAIL' that takes an 'operatie' parameter of type varchar. It uses a cursor to read from a file named 'data_phone.txt' located in 'FILE_DIR'. The procedure then updates the 'nr_calatori' column in the 'extra_pack.varchar_table' table based on the value of 'operatie'. The 'Script Output' pane shows the successful compilation of the procedure.

```
create or replace procedure update_phone_email(operatie in varchar, nr_calatori_actualizati out nocopy
is -- ultimul atribut este nocopy pentru ca procedura sa imi puna valoarea in aceasta variabila chiar
-- o eroare (raise_application_error)
file_handle utl_file.file_type;
file varchar(100);
type indexed_table is table of number index by pls_integer;
hash_table indexed_table;
words extra_pack.varchar_table := extra_pack.varchar_table();
v_email vector_email;
v_id_calator number;
begin
    if lower(operatie) = 'update_phone' then
        file_handle := utl_file.open('FILE_DIR', 'data_phone.txt', 'r');
        loop
            begin
                utl_file.get_line(file_handle, file);
                if file = '' then
                    exit;
                end;
                words := tokenize(file);
                v_email := vector_email(words);
                v_id_calator := number(v_email);
                update_phone_email('update_phone', v_id_calator);
            end;
        end;
    else
        raise_application_error(-20000, 'Operatie invalida');
    end if;
end;
/
declare
    nr_return number;
begin
    update_phone_email('update_phone', nr_return);
    dbms_output.put_line('S-au actualizat ' || nr_return || ' inregistrari');
exception
    when exceptii.operatie_invalida then
        dbms_output.put_line(SQLERRM);
        dbms_output.put_line('Incercat o operatie permisa: "update_phone" / "update_email"');
    when exceptii.limita_email_depasita then
        dbms_output.put_line(SQLERRM);
        dbms_output.put_line('S-au actualizat doar' || nr_return || ' inregistrari');
end;
/
rollback;
-- 7.
```

The screenshot shows the continuation of the 'update_phone_email' procedure. It includes a declaration section with a cursor 'nr_return' of type number. The main body of the procedure calls the function 'update_phone_email' with the argument 'update_phone' and stores the result in 'nr_return'. It then outputs the number of updated records using 'dbms_output.put_line'. The 'Script Output' pane shows the successful completion of the procedure.

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

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

CERINTA: Pentru toți conținutul cu o valoare a politiei mai mare sau egală decât una data să se afiseze numele, prenumele, salariul și top-ul primelor 5 trenuri (număr, an

fabricatie) in care acestia au lucrat. Ordinea in care vor aparea trenurile in top va fi data de anul fabricatiei acestora (desc). Daca doua trenuri sunt fabricate in acelasi an, acestea vor ocupa aceeasi pozitie in top-ul respectiv. Daca nu exista date despre faptul ca un controlor ar fi lucrat in vreun tren, atunci afisati un mesaj corespunzator in loc de top-ul trenurilor. Daca un controlor nu a lucrat in suficiente trenuri astfel incat sa se intocmeasca un top 5 pentru acesta, se va afisa top-ul pana la pozitia la care se poate ajunge cu datele din baza de date.

La final afisati si numarul controlorilor care au lucrat in cel putin un tren, dar si media salariilor acestora.

COD:

```

create or replace procedure afis_info_controlori_politete(min_politete in number) is
cursor expr_curs(polit number) is
select s.nume, s.prenume, s.salariu, cursor (
    select t.nr_tren, t.an_fabricatie
    from lucreaza_in l
    join trenuri t on l.nr_tren = t.nr_tren
    where l.id_membru = s.id_membru
    order by an_fabricatie desc
)
from controlori c
join staff s on c.id_membru = s.id_membru
where c.politete >= polit;

v_cursor sys_refcursor;
v_nume varchar(50);
v_prenume varchar(50);
v_salariu number;
v_nr_tren number;
v_an_fabricatie number;
counter_lucrat number;
top_counter number;
prev_an_fabricatie number(4);
suma_salariu number;
begin
open expr_curs(min_politete);
counter_lucrat := 0;
suma_salariu := 0;
loop
    fetch expr_curs into v_nume, v_prenume, v_salariu, v_cursor;

```

```

exit when expr_curs%notfound;

dbms_output.put_line(v_num || ' ' || v_prenume || '-' || v_salariu);
dbms_output.put_line('-----');

top_counter := 0;
prev_an_fabricatie := 0;
loop
  fetch v_cursor into v_nr_tren, v_an_fabricatie;
  if top_counter = 0 and v_cursor%notfound then
    dbms_output.put_line('Acest controlor nu a lucrat in nici un tren');
    exit;
  end if;
  exit when v_cursor%notfound;

  if v_an_fabricatie <> prev_an_fabricatie then
    top_counter := top_counter + 1;
  end if;

  dbms_output.put_line(top_counter || '. NR: ' || v_nr_tren || ' -> AN:' || v_an_fabricatie);
  prev_an_fabricatie := v_an_fabricatie;
end loop;
if top_counter > 0 then
  counter_lucrat := counter_lucrat + 1;
  suma_salariu := suma_salariu + v_salariu;
end if;

dbms_output.new_line;
dbms_output.new_line;

end loop;

if counter_lucrat > 0 then
  dbms_output.put_line(counter_lucrat || ' controlorii cu politete mai mare sau egala decat '
|| min_politete || ' de au lucrat in cel putin un tren');
  dbms_output.put_line('Media salariilor acestora este: ' || round(suma_salariu /
counter_lucrat, 2));
  else
    dbms_output.put_line('Nici un controlor cu politete de cel putin ' || min_politete || ' nu a
lucrat in vreun tren pana acum');
  end if;

```

end;

The screenshot shows the Oracle SQL Developer interface. In the left sidebar, under 'Connections', there is a tree view of database objects including tables, procedures, and packages. The main workspace shows a query builder window with the following PL/SQL code:

```
create or replace procedure afis_info_controlori_politeste(min_politeste in number) is
    cursor expr_curs(polit number) is
        select s.nume, s.prenume, s.salariu, cursor (
            select t.nr_tren, t.an_fabricatie
            from luceaza_in l
            join trenuri t on l.nr_tren = t.nr_tren
            where l.id_membru = s.id_membru
            order by an_fabricatie desc
        )
        from controlorii c
        join staff s on c.id_membru = s.id_membru
        where c.politeste >= polit;
    v_cursor sys_refcursor;
    v_nume varchar(50);
    v_prenume varchar(50);
    v_salariu number;
    v_nr_tren number;
    v_an_fabricatie number;

```

Below the code, the message 'Procedure AFIS_INFO_CONTROLORI_POLITESTE compiled' is displayed. The status bar at the bottom indicates the file is saved at 'C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql'.

APEL:

execute afis_info_controlori_politeste(2);

The screenshot shows the Oracle SQL Developer interface. The workspace displays the same PL/SQL code as before, but the 'Script Output' tab at the bottom shows the results of the execution:

```
Sava Alexandru - 5500
-----
1. NR: 28 -> AN:2011

Popescu Madalina-Andreea - 7800
-----
1. NR: 75 -> AN:2017
1. NR: 31 -> AN:2017

Tomescu Sebastian Emil - 9000
-----
Acest controlor nu a lucrat in nici un tren

2 controlorii cu politeste mai mare sau egala decat 2 de au lucrat in
Media salariilor acestora este: 6650

```

The status bar at the bottom indicates the file is modified at 'C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql'.

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

CERINTA: Determinati numele complet (nume + prenume) al calatorilor care au fost serviti de catre un chelner al carui rating este dat, cu un meniu al carui rating este dat. Rezultatul va fi un varchar ce contine numele complete separate prin virgula si spatiu. Tratati cazul in care nu exista nici un chelner cu rating-ul specificat, cel in care nu exista nici un meniu cu rating-ul specificat, cel in care cel putin unul dintre rating-uri are valori invalide, respectiv cel in care nici un calator satisface cerinta.

COD:

```

create or replace function nr_calatori_chelner_meniu(p_rating_angajat in number,
p_rating_meniu in number)
return varchar is
cursor c_chelneri is
    select id_membru from chelneri where rating = p_rating_angajat;
cursor c_meniu is
    select id_meniu from meniuri where rating = p_rating_meniu;
cursor c_principal(p_rating number, p_id_meniu number) is
    select c.nume || ' ' || c.prenume nume_complet from serveste s
    join calatori c on c.id_calator = s.id_calator
    join chelneri ch on ch.id_membru = s.id_membru
    where s.id_meniu = p_id_meniu and ch.rating = p_rating;

rating_out_of_range exception;
no_waiter exception;
no_menu exception;
no_names exception;
pragma exception_init(rating_out_of_range, -20003);
pragma exception_init(no_waiter, -20004);
pragma exception_init(no_menu, -20005);
pragma exception_init(no_names, -20006);
dummy number;
string_nume varchar(256);
empty_string boolean := true;
begin

```

```

if p_rating_angajat < 1 or p_rating_angajat > 5 or p_rating_meniu < 1 or p_rating_meniu >
5 then
    raise rating_out_of_range;
end if;

open c_chelneri;
fetch c_chelneri into dummy;

if c_chelneri%notfound then
    close c_chelneri;
    raise no_waiter;
end if;
close c_chelneri;

open c_meniu;
fetch c_meniu into dummy;
if c_meniu%notfound then
    close c_meniu;
    raise no_menu;
end if;
close c_meniu;

for rec1 in c_meniu loop
    for rec2 in c_principal(p_rating_angajat, rec1.id_meniu) loop
        if empty_string then
            string_nume := string_nume || rec2.nume_complet;
            empty_string := false;
        else
            string_nume := string_nume || ',' || rec2.nume_complet;
        end if;
    end loop;
end loop;

if empty_string then
    raise no_names;
end if;

return string_nume;
exception
when rating_out_of_range then
    dbms_output.put_line('Unul dintre rating-uri nu este intre valorile permise (1-5)');

```

```

        return '';
when no_waiter then
    dbms_output.put_line('Nu exista nici un chelner cu rating-ul ' || p_rating_angajat);
    return '';
when no_menu then
    dbms_output.put_line('Nu exista nici un meniu cu rating-ul ' || p_rating_meniu);
    return '';
when no_names then
    dbms_output.put_line('Nu exista calatori care sa fi fost servit de catre un chelner cu
rating-ul '
    || p_rating_angajat || ' cu un meniu care sa aiba rating-ul ' || p_rating_meniu);
    return '';
when others then
    dbms_output.put_line('Alta eroare');
end;

```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' tree, which includes various database objects like tables, triggers, and packages. The central pane contains the code for the function 'nr_calatori_chelner_meniu'. The right pane shows the 'Dbsns Out...' tab, which displays the message 'Function NR_CALATORI_CHELNER_MENUU compiled'. The bottom status bar indicates the file is saved.

```

CREATE OR REPLACE FUNCTION nr_calatori_chelner_meniu(p_rating_angajat IN NUMBER, p_rating_meniu IN NUMBER)
RETURN VARCHAR IS
    CURSOR c_chelneri IS
        SELECT id_membru FROM chelneri WHERE rating = p_rating_angajat;
    CURSOR c_menui IS
        SELECT id_menui FROM meniuri WHERE rating = p_rating_meniu;
    CURSOR c_principal(p_rating NUMBER, p_id_menui NUMBER) IS
        SELECT c.nume || ' ' || c.prenume nume_complet FROM serveste s
        JOIN calatori c ON c.id_calator = s.id_calator
        JOIN chelneri ch ON ch.id_membru = s.id_membru
        WHERE s.id_menui = p_id_menui AND ch.rating = p_rating;
    RATING_OUT_OF_RANGE EXCEPTION;
    NO_WAITER EXCEPTION;
    NO_MENU EXCEPTION;
    NO_NAMES EXCEPTION;
    PRAGMA EXCEPTION_INIT(RATING_OUT_OF_RANGE, -20003);
    PRAGMA EXCEPTION_INIT(NO_WAITER, -20004);
    PRAGMA EXCEPTION_INIT(NO_MENU, -20005);
    PRAGMA EXCEPTION_INIT(NO_NAMES, -20006);
    DUMMY NUMBER;

```

APEL:

```

-- cazul in care totul merge bine
begin
    dbms_output.put_line(nr_calatori_chelner_meniu(5, 4));
end;

```

```

  dbms_output.put_line('Nu exista nici un meniu cu rating-ul ' || p_rating_menu);
  return '';
when no_names then
  dbms_output.put_line('Nu exista calatori care sa fi fost servit de catre un che');
  || p_rating_angajat || ' cu un meniu care sa aiba rating-ul ' || p_rating_menu;
  return '';
when others then
  dbms_output.put_line('Alta eroare');
end;
/
-- cazul in care total merge bine
begin
  dbms_output.put_line(nr_calatori_chelner_menu(5, 4));
end;
/
-- cazul in care este apelat cu valori invalide
begin
  dbms_output.put_line(nr_calatori_chelner_menu(6, 4));
  dbms_output.put_line(nr_calatori_chelner_menu(5, 6));
end;
/

```

PL/SQL procedure successfully completed.

-- cazul in care este apelat cu valori invalide

```

begin
  -- rating-ul chelnerului este invalid
  dbms_output.put_line(nr_calatori_chelner_menu(6, 4));
  -- rating-ul meniului este invalid
  dbms_output.put_line(nr_calatori_chelner_menu(5, 6));
end;

```

```

begin
  dbms_output.put_line(nr_calatori_chelner_menu(5, 4));
end;
/
-- cazul in care este apelat cu valori invalide
begin
  -- rating-ul chelnerului este invalid
  dbms_output.put_line(nr_calatori_chelner_menu(6, 4));
  -- rating-ul meniului este invalid
  dbms_output.put_line(nr_calatori_chelner_menu(5, 6));
end;
/
-- cazul in care nu exista chelner cu rating-ul dat
begin
  dbms_output.put_line(nr_calatori_chelner_menu(3, 4));
end;
/
-- cazul in care nu exista meniu cu rating-ul dat
begin
  dbms_output.put_line(nr_calatori_chelner_menu(5, 1));
end;

```

PL/SQL procedure successfully completed.

-- cazul in care nu exista chelner cu rating-ul dat

```

begin
    dbms_output.put_line(nr_calatori_chelner_menui(3, 4));
end;

```

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

- Connections:** Oracle Connections, SGBD_Project
- Compiler Log:** Rezolvare_Cerinte.sql
- SQL Worksheet:** History, insert into menui
- Output:** SGBD_Project, Buffer Size: 20000. The output window displays the message: "Nu exista nici un chelner cu rating-ul 3".
- Script Output:** Task completed in 0.037 seconds.
- Message:** PL/SQL procedure successfully completed.

```

-- cazul in care este apelat cu valori invalide
begin
    -- rating-ul chelnerului este invalid
    dbms_output.put_line(nr_calatori_chelner_menui(6, 4));
    -- rating-ul meniuului este invalid
    dbms_output.put_line(nr_calatori_chelner_menui(5, 6));
end;
/
-- cazul in care nu exista chelner cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(3, 4));
end;
/
-- cazul in care nu exista meniu cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(5, 1));
end;
/
-- cazul in care nu exista calatori care sa satisfaca cerinta
begin
    dbms_output.put_line(nr_calatori_chelner_menui(1, 2));
end;
/
PL/SQL procedure successfully completed.

```

```

-- cazul in care nu exista meniu cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(5, 1));
end;

```

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

- Connections:** Oracle Connections, SGBD_Project
- Compiler Log:** Rezolvare_Cerinte.sql
- SQL Worksheet:** History, insert into menui
- Output:** SGBD_Project, Buffer Size: 20000. The output window displays the message: "Nu exista nici un meniu cu rating-ul 1".
- Script Output:** Task completed in 0.023 seconds.
- Message:** PL/SQL procedure successfully completed.

```

-- select c.rating, m.rating from chelneri c, meniuri m
-- minus
-- cazul in care nu exista chelner cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(5, 6));
end;
/
-- cazul in care nu exista meniu cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(3, 4));
end;
/
-- cazul in care nu exista menui cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(5, 1));
end;
/
-- cazul in care nu exista calatori care sa satisfaca cerinta
begin
    dbms_output.put_line(nr_calatori_chelner_menui(1, 2));
end;
/
PL/SQL procedure successfully completed.

```

```
-- cazul in care nu exista calatori care sa satisfaca cerinta
begin
    dbms_output.put_line(nr_calatori_chelner_menui(1, 2));
end;
```

```
-- cazul in care nu exista chelner cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(3, 4));
end;
/
-- cazul in care nu exista menu cu rating-ul dat
begin
    dbms_output.put_line(nr_calatori_chelner_menui(5, 1));
end;
/
-- cazul in care nu exista calatori care sa satisfaca cerinta
begin
    dbms_output.put_line(nr_calatori_chelner_menui(1, 2));
end;
/
--select c.rating, m.rating from chelneri c, meniuri m
--minus
--select c.rating, m.rating from serveste s
--join chelneri c on s.id_membru = c.id_membru
--join meniuri m on s.id_menu = m.id_menu;
```

PL/SQL procedure successfully completed.

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

CERINTA: Cititi dintr-un fisier, aflat intr-un director ale caror nume sunt specificate date de forma <zi_saptamana>,<distanta_minima>,<procent_marire> (cate un tuplu pe fiecare linie).

Pentru fiecare linie din fisier mariti cu procent_marire salariul angajatului care a lucrat ultima data intr-o zi a saptamanii egala cu zi_saptamana intr-un tren care circula pe o ruta cu distanta mai mare decat distanta_minima. Pentru un rand din fisier nu este permis sa avem mai multi angajati care indeplinesc conditiile respective. procent_marire trebuie sa fie un numar in intervalul [0, 1) cu doua zecimale. Tratati cazul in care nu exista informatii in baza de date care sa satisfaca o linie din fisier, cel in care exista mai multi angajati care satisfac o linie din fisier, cel in care procentul nu respecta formatul stabilit, respectiv cel in care directorul sau fisierul nu exista (sunt invalide).

COD:

```
create or replace procedure inc_slr_staff_day_route(directory_path in varchar,
                                                file_name in varchar)
is
    file_handle utl_file.file_type;
    type week_days is table of char(1) index by varchar(20);
    type rec_update is record (id_angajat number, procent number);
    type varchar_table is table of varchar(20);
    type rec_table is table of rec_update;
    t_to_update rec_table := rec_table();
    zile week_days;
    invalid_file exception;
    invalid_directory_path exception;
    invalid_percentage exception;
    pragma exception_init(invalid_directory_path, -29280);
    pragma exception_init(invalid_file, -29283);
    pragma exception_init(invalid_percentage, -20007);
    fline varchar(256);
    words extra_pack.varchar_table := extra_pack.varchar_table();
    numar_zi number;
    dist_ruta number;
    nr_linie number := 0;
    v_rec rec_update;
begin
    zile('luni') := '1';
    zile('marti') := '2';
    zile('miercuri') := '3';
    zile('joi') := '4';
    zile('vineri') := '5';
    zile('sambata') := '6';
    zile('duminica') := '7';
    file_handle := utl_file.open(upper(directory_path), file_name, 'r');
loop
    begin
        utl_file.get_line(file_handle, fline);
    exception
        when no_data_found then exit;
    end;
    nr_linie := nr_linie + 1;
```

```

words := extra_pack.spurge_string_regex(fline, '[^,]+');
numar_zi := zile(lower(words(1)));
dist_ruta := to_number(words(2));
words(3) := trim(words(3)); -- in caz ca exista spatii la finalul liniilor
if not regexp_like(words(3), '^0.\d{1,2}$') then
    raise invalid_percentage;
end if;
v_rec.procent := to_number(words(3), '0.99');

select distinct s.id_membru into v_rec.id_angajat
from staff s
join lucreaza_in l on s.id_membru = l.id_membru
join trenuri t on l.nr_tren = t.nr_tren
join bilete b on t.nr_tren = b.nr_tren
join rute r on r.id_ruta = l.id_ruta
where to_char(l.data_ultima, 'd') = numar_zi and r.distanta > dist_ruta;

t_to_update.extend;
t_to_update(t_to_update.last) := v_rec;

end loop;

forall i in t_to_update.first..t_to_update.last
    update staff set salariu = salariu * (1 + t_to_update(i).procent) where id_membru
= t_to_update(i).id_angajat;

dbms_output.put_line('S-au prelucrat cu succes cele ' || nr_linie || ' linii');
utl_file.fclose(file_handle);
exception
    when invalid_file then
        dbms_output.put_line('Fisierul "' || file_name || '" nu exista sau este invalid');
    when invalid_directory_path then
        dbms_output.put_line('Directorul "' || upper(directory_path) || '" nu exista sau este
invalid');
    when too_many_rows then
        dbms_output.put_line('Mai multi angajati care satisfac datele din fisier de la linia
' || nr_linie || '. Nepermis!');
        utl_file.fclose(file_handle);
    when no_data_found then

```

```

        dbms_output.put_line('Nu exista angajati care sa satisfaca datele din fisier de la
linia ' || nr_linie);

        utl_file.fclose(file_handle);
when invalid_percentage then
        dbms_output.put_line('Procent invalid la linia ' || nr_linie || '. Formatul permis este
"0.\d{1,2}"]');
when others then
        dbms_output.put_line('Eroare necunoscuta s-a produs la linia ' || nr_linie);
end;

```

```

create or replace procedure inc_slr_staff_day_route(directory_path in varchar,
                                                    file_name in varchar)
is
    file_handle utl_file.file_type;
    type week_days is table of char(1) index by varchar(20);
    type rec_update is record (id_angajat number, procent number);
    type varchar_table is table of varchar(20);
    type rec_table is table of rec_update;
    t_to_update rec_table := rec_table();
    zile week_days;
    invalid_file exception;
    invalid_directory_path exception;
    invalid_percentage exception;
    pragma exception_init(invalid_directory_path, -29280);
    pragma exception_init(invalid_file, -29283);
    pragma exception_init(invalid_percentage, -20007);
    fline varchar(256);
    words extra_pack.varchar_table := extra_pack.varchar_table();
    numar zi number;

```

PL/SQL procedure successfully completed.

APEL:

```

-- apel care functioneaza cum ne-am astepta
-- angajatul cu id 40 si cel cu id 35
execute inc_slr_staff_day_route('file_dir','incr_salariu_ok.txt');
select id_membru, salariu from staff;
rollback;
/
-- apel in care dau ca parametru un nume de director inexistent
execute inc_slr_staff_day_route('random', 'incr_salariu_ok.txt');
/
-- apel in care dau ca parametru un nume de fisier inexistent
execute inc_slr_staff_day_route('file_dir', 'random_file.txt');
/
-- apel pentru cazul in care fisierul contine un procent in format invalid
execute inc_slr_staff_day_route('file_dir', 'incr_salariu_proc_invalid.txt');

```

```

/
-- apel pentru cazul in care pentru o anumita linie nu exista angajati care sa satisfaca
cerinta
execute inc_slr_staff_day_route('file_dir', 'incr_salariu_no_data.txt');
/
-- apel pentru cazul in care pentru o anumita linie exista mai multi angajati care satisfac
cerinta
execute inc_slr_staff_day_route('file_dir', 'incr_salariu_too_many.txt');
/

```

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

- Connections:** BD_PROJECT, gresia151, gresia151_other, gresia251, seriu25, SGBD_Project, sys, Oracle NoSQL Conn, Database Schema Re.
- Compiler - Log:** Rezolvare_Cerinte.sql
- Worksheet:** Query Builder


```
-- angajatul cu id 40 si cel cu id 35
execute inc_slr_staff_day_route('file_dir','incr_salariu_ok.txt');
select id_membru, salariu from staff;
rollback;
/
-- apel in care dau ca parametru un nume de director inexistent
execute inc_slr_staff_day_route('random', 'incr_salariu_ok.txt');
/
-- apel in care dau ca parametru un nume de fisier inexistent
execute inc_slr_staff_day_route('file_dir', 'random_file.txt');
/
-- apel pentru cazul in care fisierul contine un procent in format invalid
execute inc_slr_staff_day_route('file_dir', 'incr_salariu_proc_invalid.txt');
/
-- apel pentru cazul in care pentru o anumita linie nu exista angajati care sa
cerinta
execute inc_slr_staff_day_route('file_dir', 'incr_salariu_no_data.txt');
/
-- apel pentru cazul in care pentru o anumita linie exista mai multi angajati
execute inc_slr_staff_day_route('file_dir', 'incr_salariu_too_many.txt');
/
```
- Output:** Shows the results of the executed queries, including errors and success messages.
- Script Output:** Task completed in 0.065 seconds. PL/SQL procedure successfully completed.

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

&&

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

Doi triggeri care sa implementeze constrangerea ca fiecare calator sa aiba un numar de telefon (sa nu fie null), iar in baza de date sa nu existe doua numere de telefon identice. (+ un trigger in plus pentru a goli hash_table-ul din pachetul ajutator)

Am evitat astfel mutating table.

COD:

```
create or replace package solve_mutating is
    type hash_table is table of number index by varchar(50);
    used hash_table;
end;
```

```

-- 10 & 11
-- Doi triggeri care sa implementeze constrangerea ca fiecare calator sa aiba un numar de telefon (sa nu fie null),
-- iar in baza de date sa nu existe doua numere de telefon identice.
-- Am evitat astfel mutating table.

create or replace package solve_mutating is
    type hash_table is table of number index by varchar(50);
    used hash_table;
end;
/
create or replace trigger valid_phone_comanda
before insert or update on calatori
declare
    cursor c is
        select numere_telefon as nr_tel from calatori;
begin
    for rec in c loop
        for i in rec.nr_tel.first..rec.nr_tel.last loop
            solve_mutating.used(rec.nr_tel(i)) := 1;
        end loop;
    end loop;
end;
/
Package SOLVE_MUTATING compiled

```

TRIGGER-ul la nivel comanda (BEFORE):

```
create or replace trigger valid_phone_comanda
before insert or update on calatori
declare
    cursor c is
        select numere_telefon as nr_tel from calatori;
begin
    for rec in c loop
        for i in rec.nr_tel.first..rec.nr_tel.last loop
            solve_mutating.used(rec.nr_tel(i)) := 1;
        end loop;
    end loop;
```

```

end loop;
end;

```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema structure under 'Connections' and 'SGBD Project'. The central workspace shows a 'Query Builder' window with the following PL/SQL code:

```

995    used hash_table;
996  end;
997 /
998 create or replace trigger valid_phone_comanda
999 before insert or update on calatori
1000 declare
1001   cursor c is
1002     select numere_telefon as nr_tel from calatori;
1003 begin
1004   for rec in c loop
1005     for i in rec.nr_tel.first..rec.nr_tel.last loop
1006       solve_mutating.used(rec.nr_tel(i)) := 1;
1007     end loop;
1008   end loop;
1009 end;
1010 /
1011 /
1012
1013 create or replace trigger valid_phone_linie
1014 before insert or update on calatori

```

The status bar at the bottom indicates 'Trigger VALID_PHONE_COMANDA compiled'.

TRIGGER-ul la nivel linie (BEFORE):

```

create or replace trigger valid_phone_linie
before insert or update on calatori
for each row
declare
  type hash_table is table of number index by varchar(50);
  used_local hash_table;
  iterator varchar(50);
begin
  if :new.numere_telefon is null then
    raise_application_error(-20008, 'Fiecare calator trebuie sa aiba asociat cel putin un
numar de telefon');
  end if;

  for i in :new.numere_telefon.first..:new.numere_telefon.last loop
    if used_local.exists(:new.numere_telefon(i)) then
      raise_application_error(-20010, 'Un calator nu poate sa aiba asociat acelasi numar
de telefon de mai multe ori: ' || :new.numere_telefon(i));
    end if;

```

```

if solve_mutating.used.exists(:new.numere_telefon(i)) then
    raise_application_error(-20009, 'Nu pot exista mai multi calatori care sa aiba
asociat acelasi numar de telefon: ' || :new.numere_telefon(i));
end if;

used_local(:new.numere_telefon(i)) := 1;
end loop;

iterator := used_local.first;
loop
    exit when iterator is null;
    solve_mutating.used(iterator) := 1;
    iterator := used_local.next(iterator);
end loop;
end;

```

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

- Project:** Rezolvare_Cerinte.apf
- Trigger Name:** VALID_PHONE_LINE
- Code:**

```

1011 /
1012 create or replace trigger valid_phone_linie
1013 before insert or update on calatori
1014 for each row
1015 declare
1016     type hash_table is table of number index by varchar(50);
1017     used_local hash_table;
1018     iterator varchar(50);
1019 begin
1020     if :new.numere_telefon is null then
1021         raise_application_error(-20008, 'Fiecare calator trebuie sa aiba asociat cel putin un numar de telefon');
1022     end if;
1023
1024     if updating then
1025         for i in :old.numere_telefon.first..:old.numere_telefon.last loop
1026             solve_mutating.used.delete(:old.numere_telefon(i));
1027         end loop;
1028     end if;
1029
1030     for i in :new.numere_telefon.first..:new.numere_telefon.last loop
1031

```
- Output:** Trigger VALID_PHONE_LINE compiled

TRIGGER-ul bonus pentru stergere (AFTER COMANDA):

```

create or replace trigger valid_phone_after
after insert or update on calatori

```

```

begin
    solve_mutating.used.delete;
end;

```

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database objects under the connection 'SGBD_PROJECT'. The central workspace shows the SQL Worksheet with the following code:

```

1040    solve_mutating.used(iterator) := 1;
1041    iterator := used_local.next(iterator);
1042  end loop;
1043 end;
1044 /
1045 create or replace trigger valid_phone_after
1046 after insert or update on calatori
1047 begin
1048     solve_mutating.used.delete;
1049 end;
1050 /
1051
-- Nu apare eroarea table mutating, chiar daca apelam insert-ul sub forma "insert into ... select ..."
-- Cazul in care numarul de telefon este deja existent in baza de date
1052 insert into calatori
1053 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0712345719')
1054 from dual;
1055
-- Cazul in care numarul de telefon se repeta in tabelul imbucat pe care dorim sa il inseram
1056 insert into calatori
1057
1058
Trigger VALID_PHONE_AFTER compiled

```

The code is being compiled, as indicated by the message 'Trigger VALID_PHONE_AFTER compiled' at the bottom of the worksheet.

DECLANSARE:

-- Cazul in care numarul de telefon este deja existent in baza de date
 insert into calatori
 select 100, 'TestFirstName', 'TestLastName', 20, 'M',
 vector_email('something@email.com'), tabel_telefon('0712345719')
 from dual;

Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql

```

1048     solve_mutating.used.delete;
1049 end;
1050 /
1051
1052 -- Nu apare eroarea table mutating, chiar daca apelam insert-ul sub forma "insert into ... select ..."
1053 -- Cazul in care numarul de telefon este deja existent in baza de date
1054 insert into calatori
1055 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0712345719')
1056 from dual;
1057
1058 -- Cazul in care numarul de telefon se repeta in tabelul imbricat pe care dorim sa il inseram
1059 insert into calatori

```

Error starting at line : 1,054 in command -
 insert into calatori
 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0712345719')
 from dual
 Error report -
 ORA-20009: Nu pot exista mai multi calatori care sa aiba asociat acelasi numar de telefon: 0712345719
 ORA-06512: at "SGBD_PROJECT.VALID_PHONE_LINE", line 16
 ORA-04088: error during execution of trigger 'SGBD_PROJECT.VALID_PHONE_LINE'

-- Cazul in care numarul de telefon se repeta in tabelul imbricat pe care dorim sa il inseram
 insert into calatori
 select 100, 'TestFirstName', 'TestLastName', 20, 'M',
 vector_email('something@email.com'), tabel_telefon('0765741965', '07643519864',
 '0765741965')
 from dual;

Oracle SQL Developer : C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql

```

1054 insert into calatori
1055 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0712345719')
1056 from dual;
1057
1058 -- Cazul in care numarul de telefon se repeta in tabelul imbricat pe care dorim sa il inseram
1059 insert into calatori
1060 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0765741965', '07643519864')
1061 from dual;
1062
1063 -- Cazul in care tabelul imbricat este atomic null
1064 insert into calatori
1065 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), null
1066 from dual;

```

Error starting at line : 1,059 in command -
 insert into calatori
 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0765741965', '07643519864')
 from dual
 Error report -
 ORA-20010: Un calator nu poate sa aiba asociat acelasi numar de telefon de mai multe ori: 0765741965
 ORA-06512: at "SGBD_PROJECT.VALID_PHONE_LINE", line 12
 ORA-04088: error during execution of trigger 'SGBD_PROJECT.VALID_PHONE_LINE'

```
-- Cazul in care tabelul imbricat este atomic null
insert into calatori
select 100, 'TestFirstName', 'TestLastName', 20, 'M',
vector_email('something@email.com'), null
from dual;
```

The screenshot shows the Oracle SQL Developer interface. In the center is a 'SQL Worksheet' window titled 'Rezolvare_Cerinte.sql'. The code in the worksheet is:

```
1060 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0765741965'),
1061 from dual;
1062
1063 -- Cazul in care tabelul imbricat este atomic null
1064 insert into calatori
1065 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), null
1066 from dual;
1067
1068 -- Cazul in care totul functioneaza cum trebuie (datele respecta regulile constrangerii impuse de catre trigger)
1069 insert into calatori
1070 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0765741965')
1071 from dual;
1072
```

Below the code, the 'Script Output' tab shows the error message:

```
Error starting at line : 1,064 in command -
insert into calatori
select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), null
from dual
Error report -
ORA-20008: Fiecare calator trebuie sa aiba asociat cel putin un numar de telefon
ORA-06512: at "SGBD_PROJECT.VALID_PHONE_LINIE", line 7
ORA-04088: error during execution of trigger 'SGBD_PROJECT.VALID_PHONE_LINIE'
```

```
-- Cazul in care totul functioneaza cum ar trebui (datele respecta regulile constrangerii
impuse de catre trigger => trigger-ii nu sunt declansati)
insert into calatori
select 100, 'TestFirstName', 'TestLastName', 20, 'M',
vector_email('something@email.com'), tabel_telefon('0765741965')
from dual;
```

```

1060  select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0765741965',
1061  from dual;
1062
1063 -- Cazul in care tabelul imbricat este atomic null
1064 insert into calatori
1065 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), null
1066 from dual;
1067
1068 -- Cazul in care totul functioneaza cum ar trebui (datele respecta regulile constrangerii impuse de catre trigger)
1069 insert into calatori
1070 select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), tabel_telefon('0765741965')
1071 from dual;
1072

```

Script Output: Task completed in 0.056 seconds

```

select 100, 'TestFirstName', 'TestLastName', 20, 'M', vector_email('something@email.com'), null
from dual
Error report -
ORA-20008: Fiecare calator trebuie sa aiba asociat cel putin un numar de telefon
ORA-06512: at "SGBD_PROJECT.VALID_PHONE_LINIE", line 7
ORA-04088: error during execution of trigger 'SGBD_PROJECT.VALID_PHONE_LINIE'

1 row inserted.

```

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

Am creat tabelul SCHEMA_DDL_HISTORY pentru a tine istoricul comenzielor de tip DDL care au fost rulate (cu succes).

COD:

```

create table schema_ddl_history (
    data_comanda date,
    user_name varchar(100),
    comanda varchar(20),
    tip_object varchar(50),
    nume_object varchar(100)
);

```

```

1070:  SELECT 1UU, 'testfirstname', 'testlastname', 2U, 'M', vector_email('something@gmail.com'), tapei_telefon('0765741965')
1071:  from dual;
1072:
1073: -- 12.
1074: create table schema_ddl_history (
1075:   data_comanda date,
1076:   user_name varchar(100),
1077:   comanda varchar(20),
1078:   tip_object varchar(50),
1079:   nume_object varchar(100)
1080: );
1081:
1082: /
1083: create or replace trigger ddl_tracker
1084: after create or alter or drop on schema
1085: declare
1086:   v_user_name varchar(100) := sys.login_user;
1087:   v_comanda varchar(20) := sys.event;
1088:   v_tip_object varchar(50) := sys.dictionary_obj_type;
1089:   v_nume_object varchar(100) := sys.dictionary_obj_name;
1090: begin
1091:   dbms_output.put_line('User-ul: ' || v_user_name);
1092:   dbms_output.put_line('Comanda rulata: ' || v_comanda);
1093:   dbms_output.put_line('Tipul obiectului referit: ' || v_tip_object);
1094:   dbms_output.put_line('Numele obiectului referit: ' || v_nume_object);
1095:   dbms_output.put_line('Data rularii comenzii: ' || to_char(sysdate, 'DD.MM.YYYY'));
1096: end;

```

Table SCHEMA_DDL_HISTORY created.

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

```

v_user_name varchar(100) := sys.login_user;
v_comanda varchar(20) := sys.event;
v_tip_object varchar(50) := sys.dictionary_obj_type;
v_nume_object varchar(100) := sys.dictionary_obj_name;
begin
  dbms_output.put_line('User-ul: ' || v_user_name);
  dbms_output.put_line('Comanda rulata: ' || v_comanda);
  dbms_output.put_line('Tipul obiectului referit: ' || v_tip_object);
  dbms_output.put_line('Numele obiectului referit: ' || v_nume_object);
  dbms_output.put_line('Data rularii comenzii: ' || to_char(sysdate, 'DD.MM.YYYY'));

  insert into schema_ddl_history
    values(sysdate, v_user_name, v_comanda, v_tip_object, v_nume_object);
end;

```

The screenshot shows the Oracle SQL Developer interface. In the center, a SQL Worksheet window displays the following PL/SQL code:

```

1081 /
1082 /
1083 create or replace trigger ddl_tracker
1084 after create or alter or drop on schema
1085 declare
1086   v_user_name varchar(100) := sys.login_user;
1087   v_comanda varchar(20) := sys.sysevent;
1088   v_tip_object varchar(50) := sys.dictionary_obj_type;
1089   v_nume_object varchar(100) := sys.dictionary_obj_name;
1090 begin
1091   dbms_output.put_line('User-ul: ' || v_user_name);
1092   dbms_output.put_line('Comanda rulata: ' || v_comanda);
1093   dbms_output.put_line('Tipul obiectului referit: ' || v_tip_object);
1094   dbms_output.put_line('Numele obiectului referit: ' || v_nume_object);
1095   dbms_output.put_line('Data rularii comenzii: ' || to_char(sysdate, 'DD.MM.YYYY'));
1096
1097   insert into schema_ddl_history
1098     values(sysdate, v_user_name, v_comanda, v_tip_object, v_nume_object);
1099 end;

```

Below the code, a message indicates the trigger has been compiled:

Trigger DDL_TRACKER compiled

DECLANSARE:

```

create table test_trigger(id number);
alter table test_trigger add test_column varchar(10);
create sequence test_seq;
drop sequence test_seq;
drop table test_trigger;

```

The screenshot shows the Oracle SQL Developer interface. In the center, a SQL Worksheet window displays the following DDL statements:

```

1100 /
1101 create table test_trigger(id number);
1102 alter table test_trigger add test_column varchar(10);
1103 create sequence test_seq;
1104 drop sequence test_seq;
1105 drop table test_trigger;

```

On the right side, the Output window shows the results of the commands:

- User-ul: SGBD_PROJECT
Comanda rulata: CREATE
Tipul obiectului referit: TABLE
Numele obiectului referit: TEST_TRIGGER
Data rularii comenzii: 06.01.2023
- User-ul: SGBD_PROJECT
Comanda rulata: ALTER
Tipul obiectului referit: TABLE
Numele obiectului referit: TEST_TRIGGER
Data rularii comenzii: 06.01.2023
- User-ul: SGBD_PROJECT
Comanda rulata: CREATE
Tipul obiectului referit: SEQUENCE
Numele obiectului referit: TEST_SEQ
Data rularii comenzii: 06.01.2023
- User-ul: SGBD_PROJECT
Comanda rulata: DROP
Tipul obiectului referit: SEQUENCE
Numele obiectului referit: TEST_SEQ
Data rularii comenzii: 06.01.2023
- User-ul: SGBD_PROJECT
Comanda rulata: DROP
Tipul obiectului referit: TABLE
Numele obiectului referit: TEST_TRIGGER
Data rularii comenzii: 06.01.2023

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

- File Bar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Connections:** Oracle Connections, Oracle Database 11g, gbase151, gbase151_other.
- Current Script:** Rezolvare_Cerinte.sql
- Compiler Log:** Rezolvare_Cerinte.sql | VALID_PHONE_LINE |
- SQL Worksheet History:** Contains a query: "Q# raise_application_error".
- Worksheet:** Query Builder, showing the following PL/SQL code:

```
1104    dbms_output.put_line('Numere obiectului referit: ' || v_nume_object),
1105    dbms_output.put_line('Data rularii comenzii: ' || to_char(sysdate, 'DD.MM.YYYY'));
```

insert into schema_ddl_history
values(sysdate, v_user_name, v_comanda, v_tip_object, v_nume_object);
end;
/
create table test_trigger(id number);
alter table test_trigger add test_column varchar(10);
create sequence test_seq;
drop sequence test_seq;
drop table test_trigger;

select * from schema_ddl_history;
- Script Output:** Shows the results of the last query:

DATA_COMMAND	USER_NAME	COMANDA	TIP_OBJECT	NUME_OBJECT
1	06-JAN-23	SGBD PROJECT CREATE TABLE	TEST TRIGGER	
2	06-JAN-23	SGBD PROJECT ALTER TABLE	TEST TRIGGER	
3	06-JAN-23	SGBD PROJECT CREATE SEQUENCE TEST SEQ	SEQUENCE	TEST SEQ
4	06-JAN-23	SGBD PROJECT DROP SEQUENCE TEST SEQ	SEQUENCE	TEST SEQ
5	06-JAN-23	SGBD PROJECT DROP TABLE TEST TRIGGER	TABLE	TEST TRIGGER
- Status Bar:** Saved: C:\Users\bluth\Desktop\Project_SGBD\Rezolvare_Cerinte.sql | Line 1109 Column 1 | Insert | Modified | Windows

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

COD:

```
create or replace package project_package is
    procedure update_phone_email(operatie in varchar, nr_calatori_actualizati out nocopy
number);
    procedure afis_info_controlori_politete(min_polite in number);
    function nr_calatori_chelner_menui(p_rating_angajat in number, p_rating_menui in
number)
        return varchar;
    procedure inc_slr_staff_day_route(directory_path in varchar, file_name in varchar);
end;
```

```

1110 create sequence test_seq;
1111 drop sequence test_seq;
1112 drop table test_trigger;
1113
1114
1115 select * from schema_ddl_history;
1116
1117 -- 13.
1118 create or replace package project_package is
1119     procedure update_phone_email(operatie in varchar, nr_calatori_actualizati out nocopy number);
1120     procedure afis_info_contorolri_politeze(min_politeze in number);
1121     function nr_calatori_chelneri_menu(p_rating_angajat in number, p_rating_menu in number)
1122         return varchar;
1123     procedure inc_slr_staff_day_route(directory_path in varchar, file_name in varchar);
1124 end;
1125 /
1126 create or replace package body project_package is
1127     procedure update_phone_email(operatie in varchar, nr_calatori_actualizati out nocopy number)
1128     is -- ultimul atribut este nocopy pentru ca procedura sa imi puna valoarea in aceasta variabila chiar si daca are loc
        -- eroare (raise_application_error)
        file_handle utl_file.file_type;
        fline varchar(100);
        type indexed_table is table of number index by pls_integer;
        hash_table indexed_table;
        words extra_pack.varchar_table := extra_pack.varchar_table();
        v_email vector_email;
        v_id_calator number;
begin
    if lower(operatie) = 'update_phone' then
        file_handle := utl_file.fopen('FILE_DIR', 'data_phone.txt', 'r');
        loop
            begin
                utl_file.get_line(file_handle, fline);
            exception
                when no_data_found then exit;
            end;
    end;

```

```

create or replace package body project_package is
procedure update_phone_email(operatie in varchar, nr_calatori_actualizati out nocopy number)
is -- ultimul atribut este nocopy pentru ca procedura sa imi puna valoarea in aceasta variabila chiar si daca are loc
-- eroare (raise_application_error)
file_handle utl_file.file_type;
fline varchar(100);
type indexed_table is table of number index by pls_integer;
hash_table indexed_table;
words extra_pack.varchar_table := extra_pack.varchar_table();
v_email vector_email;
v_id_calator number;
begin
if lower(operatie) = 'update_phone' then
    file_handle := utl_file.fopen('FILE_DIR', 'data_phone.txt', 'r');
    loop
        begin
            utl_file.get_line(file_handle, fline);
        exception
            when no_data_found then exit;
        end;
    end;

```

```

words := extra_pack.sparge_string_regex(fline, '[^,]+');
v_id_calator := to_number(words(1));
if not hash_table.exists(v_id_calator) then
    hash_table(v_id_calator) := 1;
end if;

insert into table(select numere_telefon from calatori where id_calator = v_id_calator)
values(words(2));

end loop;

nr_calatori_actualizati := hash_table.count;
utl_file.fclose(file_handle);

elsif lower(operatie) = 'update_email' then
    file_handle := utl_file=fopen('FILE_DIR', 'data_email.txt', 'r');
    loop
        begin
            utl_file.get_line(file_handle, fline);
        exception
            when no_data_found then exit;
        end;

words := extra_pack.sparge_string_regex(fline, '[^,]+');
v_id_calator := to_number(words(1));
select email into v_email from calatori where id_calator = v_id_calator;
if v_email.count = 2 then
    nr_calatori_actualizati := hash_table.count;
    raise_application_error(-20002, 'Calatorul cu id-ul ' || v_id_calator || ' are deja 2
adrese de email');
    end if;

if not hash_table.exists(v_id_calator) then
    hash_table(v_id_calator) := 1;
end if;

v_email.extend;
v_email(v_email.last) := words(2);
update calatori
set email = v_email

```

```

        where id_calator = v_id_calator;

    end loop;

    nr_calatori_actualizati := hash_table.count;
    utl_file.fclose(file_handle);
else
    raise_application_error(-20001, 'Operatia "' || operatie || " nu este cunoscuta!");
end if;
end;

procedure afis_info_controlori_politete(min_politete in number)
is
cursor expr_curs(polit number) is
    select s.nume, s.prenume, s.salariu, cursor (
        select t.nr_tren, t.an_fabricatie
        from lucreaza_in l
        join trenuri t on l.nr_tren = t.nr_tren
        where l.id_membru = s.id_membru
        order by an_fabricatie desc
    )
    from controlori c
    join staff s on c.id_membru = s.id_membru
    where c.politete >= polit;

v_cursor sys_refcursor;
v_nume varchar(50);
v_prenume varchar(50);
v_salariu number;
v_nr_tren number;
v_an_fabricatie number;
counter_lucrat number;
top_counter number;
prev_an_fabricatie number(4);
suma_salariu number;
begin
    open expr_curs(min_politete);
    counter_lucrat := 0;
    suma_salariu := 0;
loop
    fetch expr_curs into v_nume, v_prenume, v_salariu, v_cursor;

```

```

exit when expr_curs%notfound;

dbms_output.put_line(v_num || ' ' || v_prenume || ' - ' || v_salariu);
dbms_output.put_line('-----');

top_counter := 0;
prev_an_fabricatie := 0;
loop
    fetch v_cursor into v_nr_tren, v_an_fabricatie;
    if top_counter = 0 and v_cursor%notfound then
        dbms_output.put_line('Acest controlor nu a lucrat in nici un tren');
        exit;
    end if;
    exit when v_cursor%notfound;

    if v_an_fabricatie <> prev_an_fabricatie then
        top_counter := top_counter + 1;
    end if;

    dbms_output.put_line(top_counter || '. NR:' || v_nr_tren || ' -> AN:' ||
v_an_fabricatie);

    prev_an_fabricatie := v_an_fabricatie;
end loop;

if top_counter > 0 then
    counter_lucrat := counter_lucrat + 1;
    suma_salariu := suma_salariu + v_salariu;
end if;

dbms_output.new_line;
dbms_output.new_line;

end loop;

if counter_lucrat > 0 then
    dbms_output.put_line(counter_lucrat || ' controlori cu politete mai mare sau egala
decat ' || min_polite || ' de au lucrat in cel putin un tren');
    dbms_output.put_line('Media salariilor acestora este: ' || round(suma_salariu /
counter_lucrat, 2));
else

```

```

        dbms_output.put_line('Nici un controlor cu politete de cel putin ' || min_politete || 'nu
a lucrat in vreun tren pana acum');

        end if;
    end;

function nr_calatori_chelner_meniu(p_rating_angajat in number, p_rating_meniu in
number)
return varchar
is
cursor c_chelneri is
    select id_membru from chelneri where rating = p_rating_angajat;
cursor c_meniu is
    select id_meniu from meniuri where rating = p_rating_meniu;
cursor c_principal(p_rating number, p_id_meniu number) is
    select c.nume || ' ' || c.prenume nume_complet from serveste s
    join calatori c on c.id_calator = s.id_calator
    join chelneri ch on ch.id_membru = s.id_membru
    where s.id_meniu = p_id_meniu and ch.rating = p_rating;

rating_out_of_range exception;
no_waiter exception;
no_menu exception;
no_names exception;
pragma exception_init(rating_out_of_range, -20003);
pragma exception_init(no_waiter, -20004);
pragma exception_init(no_menu, -20005);
pragma exception_init(no_names, -20006);
dummy number;
string_nume varchar(256);
empty_string boolean := true;
begin
    if p_rating_angajat < 1 or p_rating_angajat > 5 or p_rating_meniu < 1 or
p_rating_meniu > 5 then
        raise rating_out_of_range;
    end if;

    open c_chelneri;
    fetch c_chelneri into dummy;

    if c_chelneri%notfound then
        close c_chelneri;

```

```

        raise no_waiter;
    end if;
    close c_chelneri;

    open c_menuiu;
    fetch c_menuiu into dummy;
    if c_menuiu%notfound then
        close c_menuiu;
        raise no_menu;
    end if;
    close c_menuiu;

    for rec1 in c_menuiu loop
        for rec2 in c_principal(p_rating_angajat, rec1.id_menuiu) loop
            if empty_string then
                string_nume := string_nume || rec2.nume_complet;
                empty_string := false;
            else
                string_nume := string_nume || ',' || rec2.nume_complet;
            end if;
        end loop;
    end loop;

    if empty_string then
        raise no_names;
    end if;

    return string_nume;
exception
    when rating_out_of_range then
        dbms_output.put_line('Unul dintre rating-uri nu este intre valorile permise (1-5)');
        return '';
    when no_waiter then
        dbms_output.put_line('Nu exista nici un chelner cu rating-ul ' || p_rating_angajat);
        return '';
    when no_menu then
        dbms_output.put_line('Nu exista nici un menuiu cu rating-ul ' || p_rating_menuiu);
        return '';
    when no_names then
        dbms_output.put_line('Nu exista calatori care sa fi fost serviti de catre un chelner cu
rating-ul '

```

```

|| p_rating_angajat || ' cu un meniu care sa aiba rating-ul ' || p_rating_meniu);
return ";
when others then
    dbms_output.put_line('Alta eroare');
end;

procedure inc_slr_staff_day_route(directory_path in varchar, file_name in varchar)
is
    file_handle utl_file.file_type;
    type week_days is table of char(1) index by varchar(20);
    type rec_update is record (id_angajat number, procent number);
    type varchar_table is table of varchar(20);
    type rec_table is table of rec_update;
    t_to_update rec_table := rec_table();
    zile week_days;
    invalid_file exception;
    invalid_directory_path exception;
    invalid_percentage exception;
    pragma exception_init(invalid_directory_path, -29280);
    pragma exception_init(invalid_file, -29283);
    pragma exception_init(invalid_percentage, -20007);
    fline varchar(256);
    words extra_pack.varchar_table := extra_pack.varchar_table();
    numar_zi number;
    dist_ruta number;
    nr_linie number := 0;
    v_rec rec_update;
begin
    zile('luni') := '1';
    zile('marti') := '2';
    zile('miercuri') := '3';
    zile('joi') := '4';
    zile('vineri') := '5';
    zile('sambata') := '6';
    zile('duminica') := '7';
    file_handle := utl_file.fopen(upper(directory_path), file_name, 'r');
loop
    begin
        utl_file.get_line(file_handle, fline);
    exception
        when no_data_found then exit;

```

```

end;
nr_linie := nr_linie + 1;

words := extra_pack.sparge_string_regex(fline, '[^,]+');
numar_zi := zile(lower(words(1)));
dist_ruta := to_number(words(2));
words(3) := trim(words(3)); -- in caz ca exista spatii la finalul liniilor
if not regexp_like(words(3), '^0\.\d{1,2}$') then
    raise invalid_percentage;
end if;
v_rec.procent := to_number(words(3), '0.99');

select distinct s.id_membru into v_rec.id_angajat
from staff s
join lucreaza_in l on s.id_membru = l.id_membru
join trenuri t on l.nr_tren = t.nr_tren
join bilete b on t.nr_tren = b.nr_tren
join rute r on r.id_ruta = l.id_ruta
where to_char(l.data_ultima, 'd') = numar_zi and r.distanta > dist_ruta;

t_to_update.extend;
t_to_update(t_to_update.last) := v_rec;

end loop;

forall i in t_to_update.first..t_to_update.last
    update staff set salariu = salariu * (1 + t_to_update(i).procent) where id_membru =
t_to_update(i).id_angajat;

dbms_output.put_line('S-au prelucrat cu succes cele ' || nr_linie || ' linii');
utl_file.fclose(file_handle);
exception
    when invalid_file then
        dbms_output.put_line('Fisierul "' || file_name || '" nu exista sau este invalid');
    when invalid_directory_path then
        dbms_output.put_line('Directorul "' || upper(directory_path) || '" nu exista sau este
invalid');
    when too_many_rows then
        dbms_output.put_line('Mai multi angajati care satisfac datele din fisier de la linia ' || nr_linie || '. Nepermis!');
        utl_file.fclose(file_handle);

```

```

when no_data_found then
    dbms_output.put_line('Nu exista angajati care sa satisfaca datele din fisier de la linia '
|| nr_linie);
    utl_file.fclose(file_handle);
when invalid_percentage then
    dbms_output.put_line('Procent invalid la linia ' || nr_linie || '. Formatul permis este
"0.\d{1,2}"]');
when others then
    dbms_output.put_line('Eroare necunoscuta s-a produs la linia ' || nr_linie);
end;
end;

```

```

create or replace package body project_package is
    procedure update_phone_email(operatie in varchar, nr_calatori_actualizati out nocopy number)
    is -- ultimul atribut este nocopy pentru ca procedura sa imi puna valoarea in aceasta variabila chiar si daca are loc
       -- o eroare (raise_application_error)
        file_handle utl_file.file_type;
        fline varchar(100);
        type indexed_table is table of number index by pls_integer;
        hash_table indexed_table;
        words extra_pack.varchar_table := extra_pack.varchar_table();
        v_email vector_email;
        v_id_calator number;
    begin
        if lower(operatie) = 'update_phone' then
            file_handle := utl_file.open('FILE_DIR', 'data_phone.txt', 'r');
            loop
                begin
                    utl_file.get_line(file_handle, fline);
                exception
                    when no_data_found then exit;
                end;
            end;
        end;
    end;

```

Package Body PROJECT_PACKAGE compiled

APELURI:

```

declare
    nr_return number;
begin
    project_package.update_phone_email('update_email', nr_return);
    dbms_output.put_line('S-au actualizat ' || nr_return || ' inregistrari');
exception
    when exceptii.operatie_invalida then
        dbms_output.put_line(SQLERRM);
        dbms_output.put_line('Incercati o operatie permisa: "update_phone" / "update_email"');
    when exceptii.limita_email_depasita then

```

```

dbms_output.put_line(SQLERRM);
if nr_return = 0 then
    dbms_output.put_line('Nu s-a actualizat nici o inregistrare');
elsif nr_return = 1 then
    dbms_output.put_line('S-a actualizat doar o inregistrare');
else
    dbms_output.put_line('S-au actualizat doar ' || nr_return || ' inregistrari');
end if;
end;

```

Oracle SQL Developer : C:\Users\bluth\Desktop\Proiect_SGBD\Rezolvare_Cerinte.sql

File Edit View Navigate Run Source Tools Window Help

Connections Oracle Connections SGBD_PROJECT

Rezolvare_Cerinte.sql Compiler - Log VALID_PHONE_LINE

SQL Worksheet History

Worksheet Query Builder

Script Output Buffer Size: 20000

ORA-20002: Calatorul cu id-ul 20 are doar o inregistrare

1457 -- APELURI ALE FUNCTIILOR / PROCEDURILOR DIN PACHET

1458

1459 declare

1460 nr_return number;

1461 begin

1462 project_package.update_phone_email('update_email', nr_return);

1463 dbms_output.put_line('S-au actualizat ' || nr_return || ' inregistrari');

1464 exception

1465 when exceptii.operatie_invalida then

1466 dbms_output.put_line(SQLERRM);

1467 dbms_output.put_line('Incercat o operatie permisa: "update_phone" / "update_email"');

1468 when exceptii.limita_email_depasita then

1469 dbms_output.put_line(SQLERRM);

1470 if nr_return = 0 then

1471 dbms_output.put_line('Nu s-a actualizat nici o inregistrare');

1472 elsif nr_return = 1 then

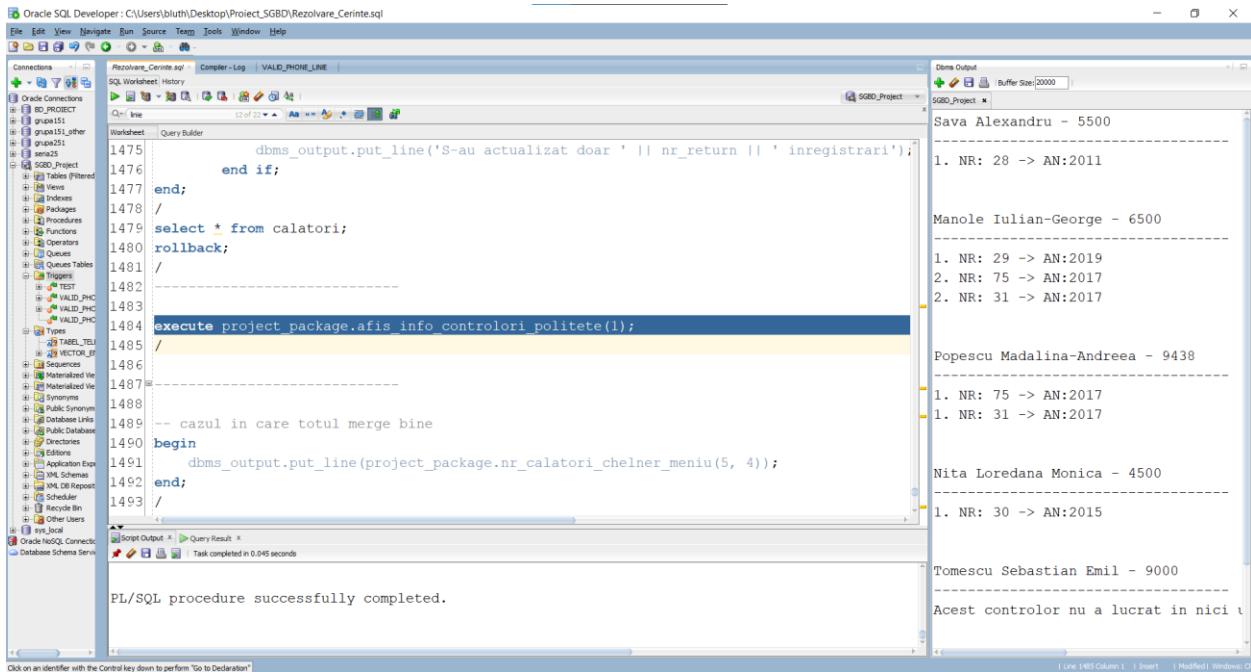
1473 dbms_output.put_line('S-a actualizat doar o inregistrare');

1474 else

1475 dbms_output.put_line('S-au actualizat doar ' || nr_return || ' inregistrari');

PL/SQL procedure successfully completed.

```
execute project_package.afis_info_controlori_politete(1);
```



```

begin
    dbms_output.put_line(project_package.nr_calatori_chelner_menui(5, 4));
end;
/
-- cazul in care este apelat cu valori invalide
begin
    -- rating-ul chelnerului este invalid
    dbms_output.put_line(project_package.nr_calatori_chelner_menui(6, 4));
    -- rating-ul meniului este invalid
    dbms_output.put_line(project_package.nr_calatori_chelner_menui(5, 6));
end;
/
-- cazul in care nu exista chelner cu rating-ul dat
begin
    dbms_output.put_line(project_package.nr_calatori_chelner_menui(3, 4));
end;
/
-- cazul in care nu exista meniu cu rating-ul dat
begin

```

```

dbms_output.put_line(project_package.nr_calatori_chelner_menui(5, 1));
end;
/
-- cazul in care nu exista calatori care sa satisfaca cerinta
begin
    dbms_output.put_line(project_package.nr_calatori_chelner_menui(1, 2));
end;

```

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' and 'Database Object Navigator' panes are visible. The main workspace contains a 'Query Builder' window with the code. Below it is a 'Script Output' tab showing 'PL/SQL procedure successfully completed.' To the right is a 'Logs Output' tab displaying several error messages:

- Constantinescu Catalin, Pavel Doina
- Unul dintre rating-uri nu este intre valoarea minima si maxima
- Nu exista nici un chelner cu rating-ul dat
- Nu exista nici un meniu cu rating-ul dat
- Nu exista calatori care sa fi fost serviti de chelnerul dat

```

execute project_package.inc_slr_staff_day_route('file_dir','incr_salariu_ok.txt');
select id_membru, salariu from staff;
rollback;
/
-- apel in care dau ca parametru un nume de director inexistent
execute project_package.inc_slr_staff_day_route('random', 'incr_salariu_ok.txt');
/
-- apel in care dau ca parametru un nume de fisier inexistent
execute project_package.inc_slr_staff_day_route('file_dir', 'random_file.txt');
/
-- apel pentru cazul in care fisierul contine un procent in format invalid
execute project_package.inc_slr_staff_day_route('file_dir', 'incr_salariu_proc_invalid.txt');
/
-- apel pentru cazul in care pentru o anumita linie nu exista angajati care sa satisfaca cerinta
execute project_package.inc_slr_staff_day_route('file_dir', 'incr_salariu_no_data.txt');
/

```

-- apel pentru cazul in care pentru o anumita linie exista mai multi angajati care satisfac cerinta

```
execute project_package.inc_slr_staff_day_route('file_dir', 'incr_salary_too_many.txt');
```

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

- Connections:** Oracle Connections, SGBD_Project
- Current File:** Rezolvare_Cerinte.sql
- Toolbars:** Standard, SQL Worksheet, History
- Left Sidebar:** Oracle Connections, SGBD_Project, Tables (Filtered), Views, Indexes, Procedures, Functions, Operators, Queues, Advanced Tables, Triggers, TEST, VALID_PHC, VALID_PHC, VALID_PHC, VALID_PHC, Types, TABLE_TEL, VECTOR_BT, BLOBs, Materialized View, Synonyms, Public Synonyms, Database Links, Public Database, Directories, Editors, Help, Application Expr, XML Schemas, XML DB Report, Scheduler, Recycle Bin, Oracle User Accounts, Local.
- Central Area:** A large code editor window containing a PL/SQL procedure named `Rezolvare_Cerinte`. The code performs various database operations like `execute project_package.inc_slr_staff_day_route` with different parameters ('file_dir', 'incr_salariu_ok.txt', 'random', 'incr_salariu_ok.txt', 'random_file.txt', etc.). It also includes a `rollback;` statement at line 1524.
- Output Window:** Shows the results of the execution:
 - PL/SQL procedure successfully completed.
 - 2 rows selected.
- Right Panel:** Shows the following messages:
 - S-a prelucrat cu succes cele 2 linii
 - Directorul "RANDOM" nu exista sau este
 - Fisierul "random_file.txt" nu exista
 - Procent invalid la linia 2. Formatul g
 - Nu exista angajati care sa satisfaca
 - Mai multi angajati care satisfac date

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

COD:

```
create or replace package extra_pack is
type indexed_numbers is table of number index by pls_integer;
type varchar_table is table of varchar(256);
type capete_rute is record(oras_plecare orase.id_oras%type, oras_sosire
orase.id_oras%type);
type table_rute is table of capete_rute;
type nr_table_rectype is record (nr_calatori number, rute table_rute);
type vec_rec is varray(2) of nr_table_rectype;
type number_rec is record (id number);
cursor c_incr_salariu return number_rec;
-- primeste ca input un string si o expresie regulata si sparge acel string in substring-uri
folosind
```

-- expresa regulata data ca parametru; returneaza un tabel imbricat cu acele substring-uri rezultate

```
function sparge_string_regex(input_string in varchar, regexp in varchar) return varchar_table;
```

-- returneaza un vector in care sunt informatii despre numarul minim, respectiv maxim de calatori

-- care au circulat pe rute, respectiv capetele rutelor respective (pot fi mai multe rute cu nr min/max de calatori)

```
function ruta_min_max return vec_rec;
```

-- procedura care maresti salariul cu 5% primilor 3 angajati de luna aceasta pentru fiecare job

-- top-ul este dat de numarul de "task-uri" efectuate (serveste, controleaza, gateste)

```
procedure incr_salary_top3;
```

```
salarii_deja_incr exception;
```

```
pragma exception_init(salarii_deja_incr, -20011);
```

-- seteaza ultima luna pentru procedura anterioara; doar pentru testing...

```
procedure set_luna_ultima_inc(luna number);
```

-- procedura care afiseaza pentru fiecare tip de reducere varsta calatorilor care au beneficiat

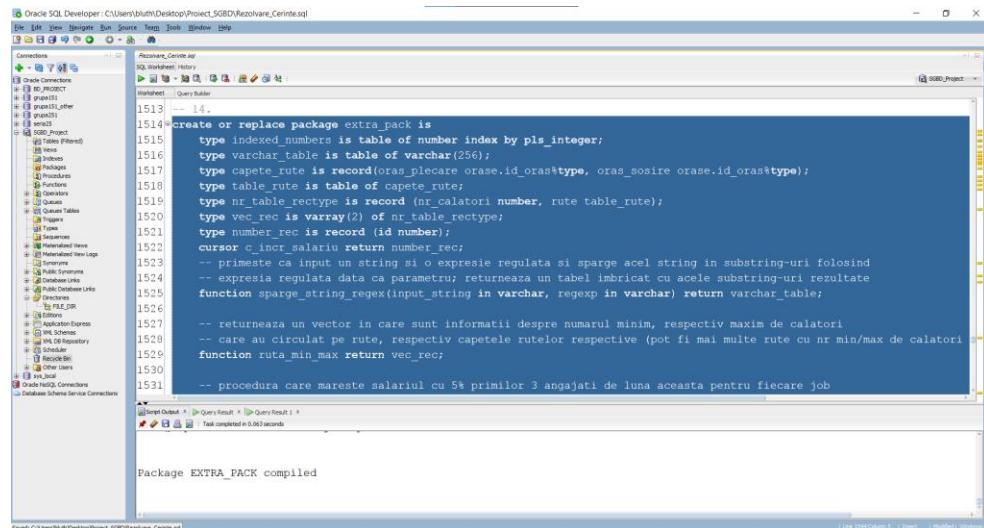
-- de acel tip de reducere (valori distincte), media varstelor si suma de bani pierduta pentru acel

-- tip de reducere; la final afiseaza suma totala pierduta pentru toate reducerile

```
procedure bani_pierduti_reduceri;
```

```
end;
```

```
/
```



The screenshot shows the Oracle SQL Developer interface with the code editor open. The code is a PL/SQL package named 'EXTRA_PACK'. The package contains several declarations and two procedures: 'incr_salary_top3' and 'bani_pierduti_reduceri'. The code includes comments explaining the purpose of each part, such as calculating salaries for the top 3 employees of the month and calculating average ages and lost money for different discount types.

```
create or replace package extra_pack is
    type indexed_numbers is table of number index by pls_integer;
    type varchar_table is table of varchar(256);
    type capete_route is record (oras_plecare orase.id_oras#type, oras_sosire orase.id_oras#type);
    type table_route is table of capete_route;
    type nr_table_rectype is record (nr_calatori number, route table_route);
    type vec_rec is varray(2) of nr_table_rectype;
    type number_rec is record (id number);
    cursor c_incr_salariu return number_rec;
    -- primeste ca input un string si o expresie regulata si sparge acel string in substring-uri folosind
    -- expresia regulata data ca parametru; returneaza un tabel imbricat cu acele substring-uri rezultante
    function sparge_string_regex(input_string in varchar, regexp in varchar) return varchar_table;
    -- returneaza un vector in care sunt informatii despre numarul minim, respectiv maxim de calatori
    -- care au circulat pe rute, respectiv capetele rutelor respective (pot fi mai multe rute cu nr min/max de calatori)
    function ruta_min_max return vec_rec;
    -- procedura care maresti salariul cu 5% primilor 3 angajati de luna aceasta pentru fiecare job
    procedure incr_salary_top3;
    salarii_deja_incr exception;
    pragma exception_init(salarii_deja_incr, -20011);
    -- seteaza ultima luna pentru procedura anterioara; doar pentru testing...
    procedure set_luna_ultima_inc(luna number);
    -- procedura care afiseaza pentru fiecare tip de reducere varsta calatorilor care au beneficiat
    -- de acel tip de reducere (valori distincte), media varstelor si suma de bani pierduta pentru acel
    -- tip de reducere; la final afiseaza suma totala pierduta pentru toate reducerile
    procedure bani_pierduti_reduceri;
end;
/
Package EXTRA_PACK compiled
```

```

create or replace package body extra_pack is
    luna_ultima_inc number := to_number(to_char(add_months(trunc(sysdate, 'mm'), -1),
    'mm'));
    type number_table is table of number;

    cursor c_incr_salariu return number_rec is
        with temp1 as (
            select id_membru, count(*) from serveste
            group by id_membru order by 2 desc
        ),
        temp2 as (
            select id_membru, count(*) from controleaza
            group by id_membru order by 2 desc
        ),
        temp3 as (
            select id_membru, count(*) from gateste
            group by id_membru order by 2 desc
        )
        select id_membru from staff
        where id_membru in (
            select id_membru from temp1 where rownum <= 3
        ) or id_membru in (
            select id_membru from temp2 where rownum <= 3
        ) or id_membru in (
            select id_membru from temp3 where rownum <= 3
        )
        for update of salariu;

```

```

cursor c_pierderi is
    select tip_reducere, cursor (
        select b.pret_initial - b.pret_final pret_partial, c.varsta
        from bilete b
        join calatori c on c.id_calator = b.id_calator
        where b.tip_reducere = r.tip_reducere
    ) curs from reduceri r;

```

```

procedure bani_pierduti_reduceri is
    v_tip_reducere reduceri.tip_reducere%type;
    hash_table indexed_numbers;
    v_curs sys_refcursor;
    suma_part number;

```

```

suma_tot number := 0;
nr_part number;
suma_varsta number;
v_pret number;
v_varsta number;
iterator number;
begin
  open c_pierderi;
  loop
    fetch c_pierderi into v_tip_reducere, v_curs;
    exit when c_pierderi%notfound;
    suma_part := 0;
    nr_part := 0;
    suma_varsta := 0;

    loop
      fetch v_curs into v_pret, v_varsta;
      exit when v_curs%notfound;
      suma_varsta := suma_varsta + v_varsta;
      suma_part := suma_part + v_pret;
      nr_part := nr_part + 1;
      if not hash_table.exists(v_varsta) then
        hash_table(v_varsta) := 1;
      end if;
    end loop;

    dbms_output.put_line('===== ' || v_tip_reducere || ' =====');
    if nr_part = 0 then
      dbms_output.put_line('Nimeni nu a beneficiat de acest tip de reducere');
      dbms_output.new_line;
    else
      iterator := hash_table.first;
      loop
        exit when iterator is null;
        dbms_output.put_line(iterator || ' ani');
        iterator := hash_table.next(iterator);
      end loop;
      hash_table.delete;
      dbms_output.new_line;
      dbms_output.put_line('Media varstelor: ' || suma_varsta / nr_part || ' ani');
    end if;
  end loop;
end;

```

```

        dbms_output.put_line('Suma pierduta pentru acest tip de reducere: ' || suma_part || '
RON');
        dbms_output.new_line;
        suma_tot := suma_tot + suma_part;
        end if;

    end loop;
    close c_pierderi;

    dbms_output.put_line('In total s-au pierdut: ' || suma_tot || ' RON');
end;

procedure set_luna_ultima_inc(luna number) is
begin
    luna_ultima_inc := luna;
end;

procedure incr_salary_top3 is
    v_id number;
    v_table number_table := number_table();
begin
    if to_number(to_char(sysdate, 'mm')) = luna_ultima_inc then
        raise_application_error(-20011, 'Luna aceasta s-a realizat deja incrementarea
salariala');
    end if;

    luna_ultima_inc := to_number(to_char(sysdate, 'mm'));

    open c_incr_salariu;
    loop
        fetch c_incr_salariu into v_id;
        exit when c_incr_salariu%notfound;
        update staff set salariu = 1.05 * salariu
        where current of c_incr_salariu;

        v_table.extend;
        v_table(v_table.last) := v_id;
    end loop;
    close c_incr_salariu;

    dbms_output.put_line('S-a incrementat salariul angajatilor cu id-urile: ');

```

```

for i in v_table.first..v_table.last loop
    dbms_output.put(v_table(i) || ' ');
end loop;
dbms_output.new_line;
end;

function sparge_string_regex(input_string in varchar, regexp in varchar)
return varchar_table
is
    return_collection varchar_table := varchar_table();
begin
    select
        regexp_substr (
            input_string, regexp, 1, level
        ) bulk collect into return_collection
        from dual
        connect by regexp_substr (
            input_string, regexp, 1, level
        ) is not null;

    return return_collection;
end;

function ruta_min_max return vec_rec is
    cursor c_min is
        select id_ruta, (select count(*) from bilete where id_ruta = r.id_ruta) nr_calatori
        from rute r
        where (select count(*) from bilete where id_ruta = r.id_ruta)
        = (select min(count(*)) from bilete group by id_ruta);

    cursor c_max is
        select id_ruta, (select count(*) from bilete where id_ruta = r.id_ruta) nr_calatori
        from rute r
        where (select count(*) from bilete where id_ruta = r.id_ruta)
        = (select max(count(*)) from bilete group by id_ruta);

    v_id_ruta rute.id_ruta%type;
    v_nr_calatori number;
    first_iteration boolean := true;
    return_vec vec_rec := vec_rec();

```

```

dummy_rec nr_table_rectype;
v_capete_rute capete_rute;
begin
    dummy_rec.rute := table_rute();
    return_vec.extend(2);
    return_vec(1) := dummy_rec;
    return_vec(2) := dummy_rec;
    open c_min;
    loop
        fetch c_min into v_id_ruta, v_nr_calatori;
        exit when c_min%notfound;
        if first_iteration then
            return_vec(1).nr_calatori := v_nr_calatori;
            first_iteration := false;
        end if;
        select id_oras_plecare, id_oras_sosire into v_capete_rute
        from ruta where id_ruta = v_id_ruta;

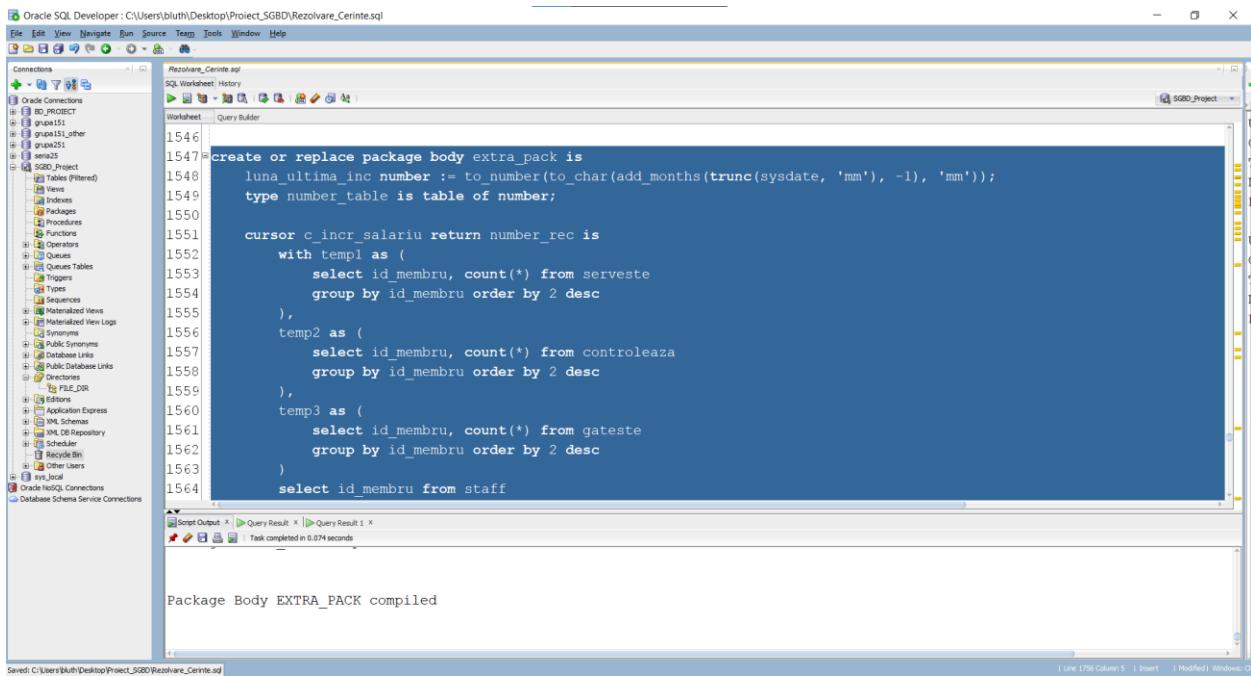
        return_vec(1).rute.extend;
        return_vec(1).rute(return_vec(1).rute.last) := v_capete_rute;
    end loop;
    close c_min;
    if first_iteration then
        raise no_data_found;
    end if;

    first_iteration := true;
    open c_max;
    loop
        fetch c_max into v_id_ruta, v_nr_calatori;
        exit when c_max%notfound;
        if first_iteration then
            return_vec(2).nr_calatori := v_nr_calatori;
            first_iteration := false;
        end if;
        select id_oras_plecare, id_oras_sosire into v_capete_rute
        from ruta where id_ruta = v_id_ruta;

        return_vec(2).rute.extend;
        return_vec(2).rute(return_vec(2).rute.last) := v_capete_rute;
    end loop;

```

```
close c_max;  
  
if first_iteration then  
    raise no_data_found;  
end if;  
  
return return_vec;  
end;  
end;
```



APELURI: (functia sparge_string_regex este folosita in exercitiile 6 si 9)
declare

```

    vec_test extra_pack.vec_rec := extra_pack.vec_rec();
begin
    vec_test := extra_pack.ruta_min_max;
    dbms_output.put_line('Nr calatori minim: ' || vec_test(1).nr_calatori || ', iar rutele sunt: ');
    for j in vec_test(1).rute.first..vec_test(1).rute.last loop
        dbms_output.put_line(vec_test(1).rute(j).oras_plecare || '->' ||
    vec_test(1).rute(j).oras_sosire);
    end loop;
    dbms_output.new_line;

    dbms_output.put_line('Nr calatori maxim: ' || vec_test(2).nr_calatori || ', iar rutele sunt: ');

```

```
for j in vec_test(2).rute.first..vec_test(2).rute.last loop
    dbms_output.put_line(vec_test(2).rute(j).oras_plecare || ' -> ' ||
vec_test(2).rute(j).oras_sosire);
end loop;
dbms_output.new_line;
exception
when no_data_found then
    dbms_output.put_line('Nu exista date');
end;
/
execute extra_pack.incr_salary_top3;
select * from staff;
--34 33 36 38 39 41 43 45 47
rollback;
/
execute extra_pack.bani_pierduti_reduceri;
/
```