GESTIUNEA INFORMAȚIILOR UNEI COMPANII DE UTILITĂȚI

1. Proiectul își propune sa proiecteze o bază de date care sa gestioneze o parte din informațiile și problemele unei companii ce oferă utilități clienților (această bază de date poate fi folosită pentru o aplicație a acestei companii, dar nu numai. Această aplicație poate fi destinată persoanelor obișnuite pentru a se înregistra, devenind astfel clienți ai companiei, pentru a-și vedea facturile, pentru a trimite plângeri în legătură cu eventuale defecțiuni, pentru a-și plăti facturile direct din aplicație).

   Regulile de funcționare ale modelului sunt:

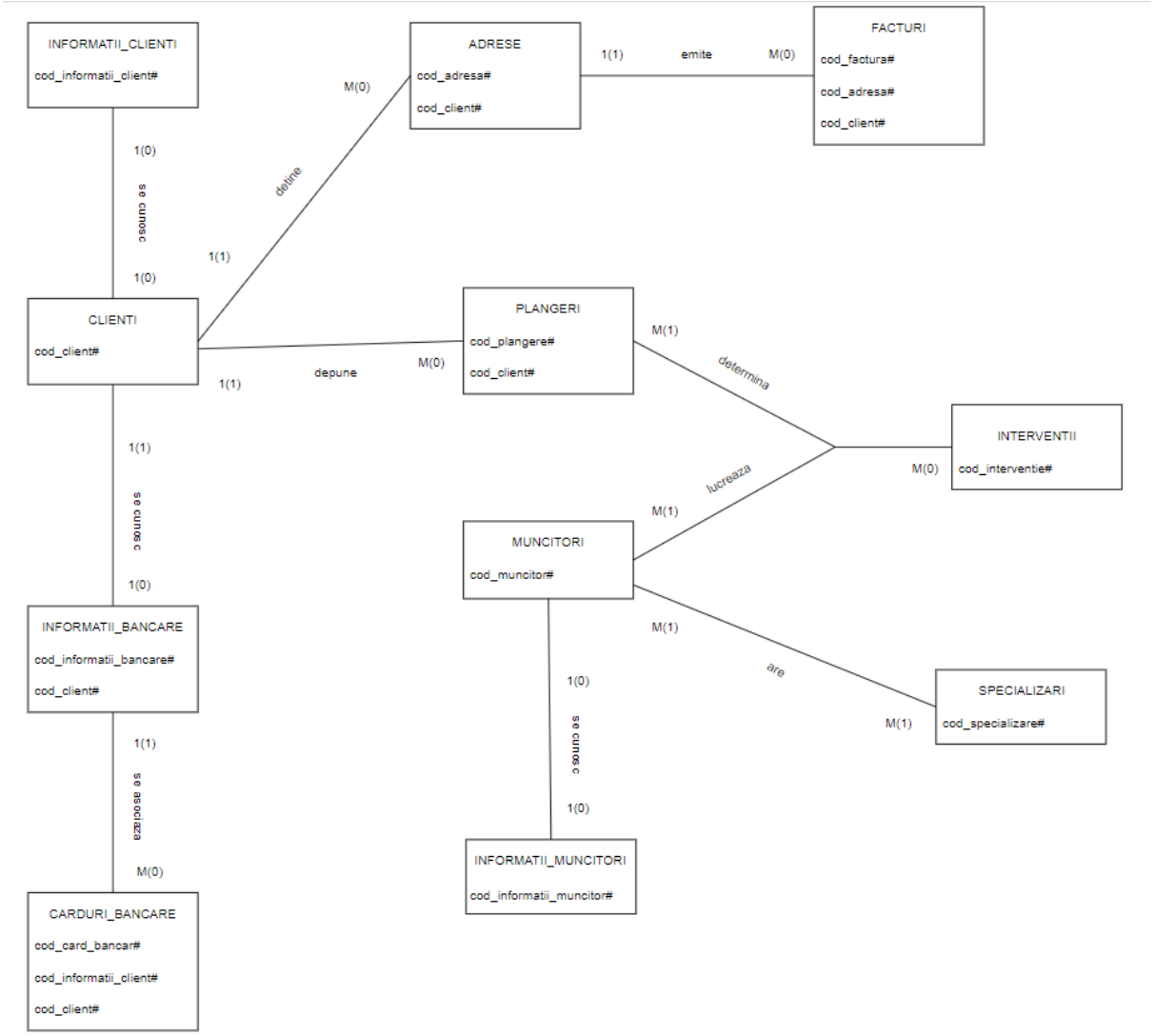
   1) Despre clienți se cunosc informații personale cat și bancare.
   2) Un client are adrese, iar pentru fiecare dintre adrese se emit facturi.
   3) Un client poate depune o plângere dacă există o defecțiune ce trebuie reparată.
   4) Dacă plângerea este considerată justificată va avea loc o intervenție pe baza acelei plângeri (sau mai multor plangeri dacă exista mai mulți clienți în aceeași zona care au depus o plângere similară), iar la intervenții vor participa muncitori (angajați ai companiei).
   5) De asemenea o singura plângere poate genera mai multe intervenții dacă se consideră că problema nu este una locală ci are loc și în cazul altor clienți.
   6) Se cunosc informații personale despre un muncitor (angajat al companiei), cât si specializările sale.

   Restricțiile impuse asupra modelului sunt:

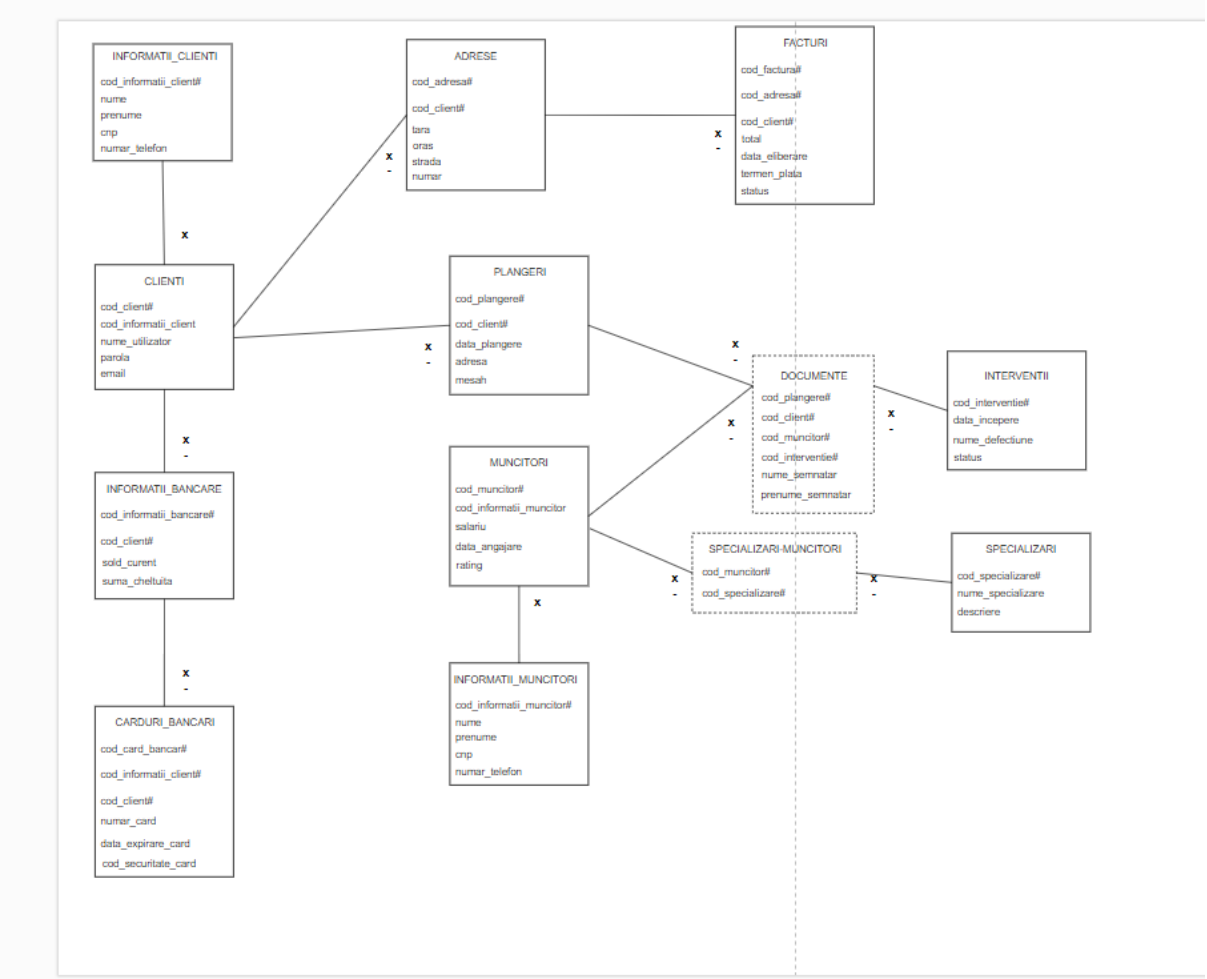
   1) Despre un client se cunosc informații din maxim un singur obiect ce oferă informații despre clienți (adică din tabelul Informatii_Clienti)
   2) Într-un obiect ce oferă informații despre clienți se cunosc informații despre cel mult un singur client.
   3) Despre un client se cunosc informații bancare din maxim un singur obiect ce oferă informații bancare despre clienți.
   4) Într-un obiect ce oferă informații bancare despre clienți se cunosc informații bancare despre un singur client.
   5) Informațiile bancare ale unui client pot avea asociate mai multe carduri bancare.
   6) Un card bancar trebuie sa fie asociat informatiilor bancare ale unui singur client.
   7) Un client poate să nu dețină nicio adresa, dar poate deține și mai multe adrese.
   8) O adresa este deținută de un singur client.
   9) La o adresă poate sa nu fie emisă nicio factură, dar pot fi emise și mai multe facturi.
   10) O factură este emisă doar la o adresă.
   11) Un client poate să nu dispuna nicio plângere, dar poate depune și mai multe plângeri.
   12) O plângere poate fi depusă de un singur client.
   13) O plângere poate să nu determine nicio intervenție, dar poate determina și mai multe.

14) Un muncitor poate sa nu participe la nicio intervenție, dar poate participa și la mai multe intervenții.
15) O intervenție este declanșată de cel puțin o plângere, dar poate fi declanșată și de mai multe plangeri.
16) La o intervenție trebuie sa participe cel puțin un muncitor, dar pot participa și mai mulți muncitori.
17) Despre un muncitor se cunosc informații din cel mult un singur obiect ce oferă informații despre muncitori.
18) Într-un obiect ce oferă informații despre muncitori (din tabelul Informatii_Muncitori) se cunosc informații despre maxim un singur muncitor.
19) Un muncitor trebuie să aibă cel puțin o specialitate, dar poate avea și mai multe.
20) Pentru fiecare specialitate există cel puțin un muncitor care o deține, dar pot exista și mai mulți.

2. Diagrama Entitate/Relatie



3. Diagrama conceptuala



4. Creare tabele:

- INFORMATII_CLIENTI:

```
CREATE TABLE INFORMATII_CLIENTI(
cod_informatii_client NUMBER(4) NOT NULL,
nume VARCHAR2(50) NOT NULL,
prenume VARCHAR2(50) NOT NULL,
cnp VARCHAR2(50) NOT NULL,
numar_telefon VARCHAR2(50) NOT NULL,
CONSTRAINT PK_Informatii_Clienti PRIMARY KEY(cod_informatii_client),
CONSTRAINT UC_Cnp UNIQUE(cnp),
CONSTRAINT UC_Numar_Telefon UNIQUE(numar_telefon));
```

```
CREATE TABLE INFORMATII_CLIENTI(
cod_informatii_client NUMBER(4) NOT NULL,
nume VARCHAR2(50) NOT NULL,
prenume VARCHAR2(50) NOT NULL,
cnp VARCHAR2(50) NOT NULL,
numar_telefon VARCHAR2(50) NOT NULL,
CONSTRAINT PK_Informatii_Clienti PRIMARY KEY(cod_informatii_client),
CONSTRAINT UC_Cnp UNIQUE(cnp),
CONSTRAINT UC_Numar_Telefon UNIQUE(numar_telefon));
```

Script Output

Task completed in 0.034 seconds

```
Table INFORMATII_CLIENTI created.
```

- CLIENTI:

```
CREATE TABLE CLIENTI(
cod_client NUMBER(4) NOT NULL,
cod_informatii_client NUMBER(4),
nume_utilizator VARCHAR2(50) NOT NULL,
parola VARCHAR2(50) NOT NULL,
email VARCHAR2(50) NOT NULL,
CONSTRAINT PK_Clienti PRIMARY KEY(cod_client),
CONSTRAINT FK_Cod_Informatii_Client FOREIGN KEY(cod_informatii_client)
REFERENCES Informatii_Clienti(cod_informatii_client),
CONSTRAINT UC_Cod_Informatii_Client UNIQUE(cod_informatii_client),
CONSTRAINT UC_Nume_Utilizator UNIQUE(nume_utilizator),
CONSTRAINT UC_Email UNIQUE(email));
```

```
CREATE TABLE CLIENTI(
cod_client NUMBER(4) NOT NULL,
cod_informatii_client NUMBER(4),
nume_utilizator VARCHAR2(50) NOT NULL,
parola VARCHAR2(50) NOT NULL,
email VARCHAR2(50) NOT NULL,
CONSTRAINT PK_Clienti PRIMARY KEY(cod_client),
CONSTRAINT FK_Cod_Informatii_Client FOREIGN KEY(cod_informatii_client) REFERENCES Informatii_Clienti(cod_informatii_client),
CONSTRAINT UC_Cod_Informatii_Client UNIQUE(cod_informatii_client),
CONSTRAINT UC_Nume_Utilizator UNIQUE(nume_utilizator),
CONSTRAINT UC_Email UNIQUE(email));
```

Script Output | Query Result

Task completed in 0.036 seconds

```
Table CLIENTI created.
```

- INFORMATII_BANCARE:

```
CREATE TABLE INFORMATII_BANCARE(
cod_informatii_bancare NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
sold_curent NUMBER(6) DEFAULT 0 NOT NULL,
suma_cheltuita NUMBER(6) DEFAULT 0 NOT NULL,
CONSTRAINT PK_Informatii_Bancare PRIMARY KEY(cod_informatii_bancare,
cod_client),
CONSTRAINT FK_Cod_Clienti FOREIGN KEY(cod_client) REFERENCES
Clienti(cod_client),
CONSTRAINT UC_Cod_Client UNIQUE(cod_client));
```

```
CREATE TABLE INFORMATII_BANCARE(
cod_informatii_bancare NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
sold_curent NUMBER(6) DEFAULT 0 NOT NULL,
suma_cheltuita NUMBER(6) DEFAULT 0 NOT NULL,
CONSTRAINT PK_Informatii_Bancare PRIMARY KEY(cod_informatii_bancare, cod_client),
CONSTRAINT FK_Cod_Clienti FOREIGN KEY(cod_client) REFERENCES Clienti(cod_client),
CONSTRAINT UC_Cod_Client UNIQUE(cod_client));
```

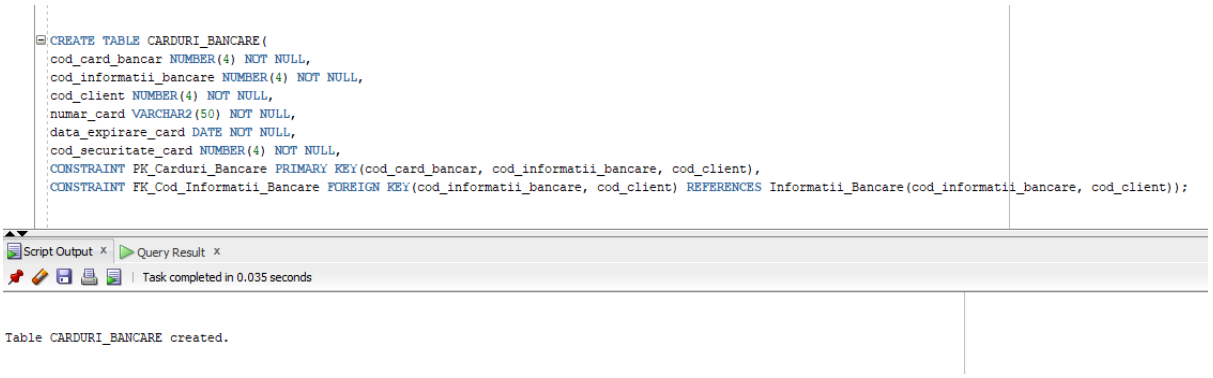
Script Output | Query Result

Task completed in 0.039 seconds

```
Table INFORMATII_BANCARE created.
```

- CARDURI_BANCARE:

```
CREATE TABLE CARDURI_BANCARE(
cod_card_bancar NUMBER(4) NOT NULL,
cod_informatii_bancare NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
numar_card VARCHAR2(50) NOT NULL,
data_expirare_card DATE NOT NULL,
cod_securitate_card NUMBER(4) NOT NULL,
CONSTRAINT PK_Carduri_Bancare PRIMARY KEY(cod_card_bancar,
cod_informatii_bancare, cod_client),
CONSTRAINT FK_Cod_Informatii_Bancare FOREIGN
KEY(cod_informatii_bancare, cod_client) REFERENCES
Informatii_Bancare(cod_informatii_bancare, cod_client));
```



- ADRESE:

```
CREATE TABLE ADRESE(
cod_adresa NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
tara VARCHAR2(50) NOT NULL,
oras VARCHAR2(50) NOT NULL,
strada VARCHAR2(50) NOT NULL,
numar NUMBER(4) NOT NULL,
CONSTRAINT PK_Adrese PRIMARY KEY(cod_adresa, cod_client),
CONSTRAINT FK_Cod_Client FOREIGN KEY(cod_client) REFERENCES
Clienti(cod_client));
```

```
CREATE TABLE ADRESE(
cod_adresa NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
tara VARCHAR2(50) NOT NULL,
oras VARCHAR2(50) NOT NULL,
strada VARCHAR2(50) NOT NULL,
numar NUMBER(4) NOT NULL,
CONSTRAINT PK_Adrese PRIMARY KEY(cod_adresa, cod_client),
CONSTRAINT FK_Cod_Client FOREIGN KEY(cod_client) REFERENCES Clienti(cod_client));
```

Script Output | Query Result

Task completed in 0.037 seconds

```
Table ADRESE created.
```

- FACTURI:

```
CREATE TABLE FACTURI(
cod_factura NUMBER(4) NOT NULL,
cod_adresa NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
total NUMBER(4) NOT NULL,
data_eliberare DATE DEFAULT TO_CHAR(SYSDATE, 'DD-MON-YYYY') NOT
NULL,
termen_plata DATE NOT NULL,
status VARCHAR(50) DEFAULT 'Neplatit' NOT NULL,
CONSTRAINT PK_Facturi PRIMARY KEY(cod_factura, cod_adresa, cod_client),
CONSTRAINT FK_Cod_Adresa FOREIGN KEY(cod_adresa, cod_client)
REFERENCES Adrese(cod_adresa, cod_client),
CONSTRAINT CH_Date CHECK (data_eliberare < termen_plata));
```

```
CREATE TABLE FACTURI(
cod_factura NUMBER(4) NOT NULL,
cod_adresa NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
total NUMBER(4) NOT NULL,
data_eliberare DATE DEFAULT TO_CHAR(SYSDATE, 'DD-MON-YYYY') NOT NULL,
termen_plata DATE NOT NULL,
status VARCHAR(50) DEFAULT 'Neplatit' NOT NULL,
CONSTRAINT PK_Facturi PRIMARY KEY(cod_factura, cod_adresa, cod_client),
CONSTRAINT FK_Cod_Adresa FOREIGN KEY(cod_adresa, cod_client) REFERENCES Adrese(cod_adresa, cod_client),
CONSTRAINT CH_Date CHECK (data_eliberare < termen_plata));
```

Script Output × Query Result ×

Task completed in 0.046 seconds

```
Table FACTURI created.
```

- PLANGERI:

```
CREATE TABLE PLANGERI(
cod_plangere NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
data_plangere DATE DEFAULT TO_CHAR(SYSDATE, 'DD-MON-YYYY') NOT NULL,
adresa VARCHAR(200) NOT NULL,
mesaj VARCHAR(2000) NOT NULL,
CONSTRAINT PK_Plangeri PRIMARY KEY(cod_plangere, cod_client),
CONSTRAINT FK_Cod_Client_for_Plangere FOREIGN KEY(cod_client)
REFERENCES Clienti(cod_client));
```

```
CREATE TABLE PLANGERI(
cod_plangere NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
data_plangere DATE DEFAULT TO_CHAR(SYSDATE, 'DD-MON-YYYY') NOT NULL,
adresa VARCHAR(200) NOT NULL,
mesaj VARCHAR(2000) NOT NULL,
CONSTRAINT PK_Plangeri PRIMARY KEY(cod_plangere, cod_client),
CONSTRAINT FK_Cod_Client_for_Plangere FOREIGN KEY(cod_client) REFERENCES Clienti(cod_client));
```

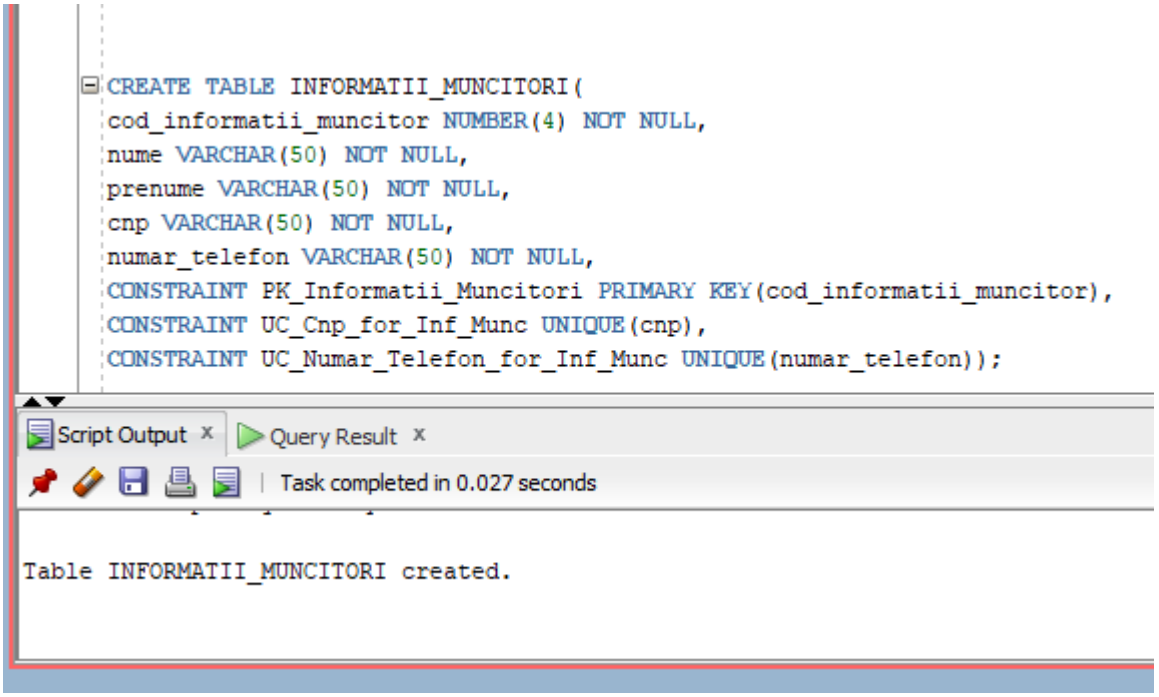
Script Output × Query Result ×

Task completed in 0.048 seconds

```
Table PLANGERI created.
```

- INFORMATII_MUNCITORI:

```
CREATE TABLE INFORMATII_MUNCITORI(
cod_informatii_muncitor NUMBER(4) NOT NULL,
nume VARCHAR(50) NOT NULL,
prenume VARCHAR(50) NOT NULL,
cnp VARCHAR(50) NOT NULL,
numar_telefon VARCHAR(50) NOT NULL,
CONSTRAINT PK_Informatii_Muncitori PRIMARY KEY(cod_informatii_muncitor),
CONSTRAINT UC_Cnp_for_Inf_Munc UNIQUE(cnp),
CONSTRAINT UC_Numar_Telefon_for_Inf_Munc UNIQUE(numar_telefon));
```



- MUNCITORI:

```
CREATE TABLE MUNCITORI(
cod_muncitor NUMBER(4) NOT NULL,
cod_informatii_muncitor NUMBER(4),
salariu NUMBER(4) NOT NULL,
data_angajare DATE DEFAULT TO_CHAR(SYSDATE, 'DD-MON-YYYY') NOT
NULL,
rating NUMBER(4),
CONSTRAINT PK_Muncitori PRIMARY KEY(cod_muncitor),
CONSTRAINT FK_Informatii_Muncitori FOREIGN KEY(cod_informatii_muncitor)
REFERENCES Informatii_Muncitori(cod_informatii_muncitor),
CONSTRAINT UC_Cod_Informatii_Muncitor UNIQUE(cod_informatii_muncitor));
```

```
CREATE TABLE MUNCITORI(
cod_muncitor NUMBER(4) NOT NULL,
cod_informatii_muncitor NUMBER(4),
salariu NUMBER(4) NOT NULL,
data_angajare DATE DEFAULT TO_CHAR(SYSDATE, 'DD-MON-YYYY') NOT NULL,
rating NUMBER(4),
CONSTRAINT PK_Muncitori PRIMARY KEY(cod_muncitor),
CONSTRAINT FK_Informatii_Muncitori FOREIGN KEY(cod_informatii_muncitor) REFERENCES Informatii_Muncitori(cod_informatii_muncitor),
CONSTRAINT UC_Cod_Informatii_Muncitor UNIQUE(cod_informatii_muncitor));
```

```
Script Output   Query Result
Task completed in 0.046 seconds

Table MUNCITORI created.
```

- SPECIALIZARI:

CREATE TABLE SPECIALIZARI(
cod_specializare NUMBER(4) NOT NULL,
nume_specializare VARCHAR2(50) NOT NULL,
descriere VARCHAR2(2000) NOT NULL,
CONSTRAINT PK_Specializari PRIMARY KEY(cod_specializare),
CONSTRAINT UC_Nume_Specializare UNIQUE(nume_specializare));

```
CREATE TABLE SPECIALIZARI(
cod_specializare NUMBER(4) NOT NULL,
nume_specializare VARCHAR2(50) NOT NULL,
descriere VARCHAR2(2000) NOT NULL,
CONSTRAINT PK_Specializari PRIMARY KEY(cod_specializare),
CONSTRAINT UC_Nume_Specializare UNIQUE(nume_specializare));
```

```
Script Output   Query Result
Task completed in 0.041 seconds

Table SPECIALIZARI created.
```

- SPECIALIZARI_MUNCITORI:

CREATE TABLE SPECIALIZARI_MUNCITORI(
cod_muncitor NUMBER(4) NOT NULL,
cod_specializare NUMBER(4) NOT NULL,
CONSTRAINT PK_Specializari_Muncitori PRIMARY KEY(cod_muncitor,
cod_specializare),

CONSTRAINT FK_Cod_Muncitor FOREIGN KEY(cod_muncitor) REFERENCES Muncitori(cod_muncitor),
CONSTRAINT FK_Cod_Specializare FOREIGN KEY(cod_specializare) REFERENCES Specializari(cod_specializare));

```
CREATE TABLE SPECIALIZARI_MUNCITORI(
cod_muncitor NUMBER(4) NOT NULL,
cod_specializare NUMBER(4) NOT NULL,
CONSTRAINT PK_Specializari_Muncitori PRIMARY KEY(cod_muncitor, cod_specializare),
CONSTRAINT FK_Cod_Muncitor FOREIGN KEY(cod_muncitor) REFERENCES Muncitori(cod_muncitor),
CONSTRAINT FK_Cod_Specializare FOREIGN KEY(cod_specializare) REFERENCES Specializari(cod_specializare));
```

Script Output | Query Result

Task completed in 0.048 seconds

```
Table SPECIALIZARI_MUNCITORI created.
```

- INTERVENTII:

CREATE TABLE INTERVENTII(
cod_interventie NUMBER(4) NOT NULL,
data_incepere DATE NOT NULL,
nume_defectiune VARCHAR(50) NOT NULL,
status VARCHAR(50) DEFAULT 'Neinceputa' NOT NULL,
CONSTRAINT PK_Interventii PRIMARY KEY(cod_interventie));

```
CREATE TABLE INTERVENTII(
cod_interventie NUMBER(4) NOT NULL,
data_incepere DATE NOT NULL,
nume_defectiune VARCHAR(50) NOT NULL,
status VARCHAR(50) DEFAULT 'Neinceputa' NOT NULL,
CONSTRAINT PK_Interventii PRIMARY KEY(cod_interventie));
```

Script Output | Query Result

Task completed in 0.048 seconds

```
Table INTERVENTII created.
```

- DOCUMENTE:

CREATE TABLE DOCUMENTE(
cod_plangere NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
cod_muncitor NUMBER(4) NOT NULL,
cod_interventie NUMBER(4) NOT NULL,
nume_semnatar VARCHAR(50) NOT NULL,
prenume_semnatar VARCHAR(50) NOT NULL,
CONSTRAINT PK_Documente PRIMARY KEY(cod_plangere, cod_client, cod_muncitor, cod_interventie),
CONSTRAINT FK_Cod_Plangere FOREIGN KEY(cod_plangere, cod_client) REFERENCES Plangeri(cod_plangere, cod_client),
CONSTRAINT FK_Cod_Muncitor_for_Documente FOREIGN KEY(cod_muncitor) REFERENCES Muncitori(cod_muncitor),
CONSTRAINT FK_Cod_Interventie FOREIGN KEY(cod_interventie) REFERENCES Interventii(cod_interventie));

```
CREATE TABLE DOCUMENTE(
cod_plangere NUMBER(4) NOT NULL,
cod_client NUMBER(4) NOT NULL,
cod_muncitor NUMBER(4) NOT NULL,
cod_interventie NUMBER(4) NOT NULL,
nume_semnatar VARCHAR(50) NOT NULL,
prenume_semnatar VARCHAR(50) NOT NULL,
CONSTRAINT PK_Documente PRIMARY KEY(cod_plangere, cod_client, cod_muncitor, cod_interventie),
CONSTRAINT FK_Cod_Plangere FOREIGN KEY(cod_plangere, cod_client) REFERENCES Plangeri(cod_plangere, cod_client),
CONSTRAINT FK_Cod_Muncitor_for_Documente FOREIGN KEY(cod_muncitor) REFERENCES Muncitori(cod_muncitor),
CONSTRAINT FK_Cod_Interventie FOREIGN KEY(cod_interventie) REFERENCES Interventii(cod_interventie));
```

Script Output | Query Result

Task completed in 0.667 seconds

```
Table DOCUMENTE created.
```

5. Inserare in tabele:

- INFORMATII_CLIENTI

INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Marian', 'Andrei', '5020413150634', '0734135426');
INSERT INTO Informatii_Clienti

```
VALUES (Secventa.nextval, 'Ion', 'Vasile', '5026262782345', '0774357120');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ioana', 'Cristina', '5022494157062', '0732584214');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Buzatu', 'Ionut', '5021395602456', '0712690369');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ana', 'Cosmina', '5021049258568', '0725790632');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Gigel', 'Frone', '5021256073548', '0724679520');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Andreea', 'Maria', '5021249056732', '0724905687');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Cristi', 'Sebastian', '5042458037893', '0738952469');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Dragos', 'Andrei', '5015730458972', '0739503182');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ionut', 'Dima', '5072490567120', '0719023567');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ion', 'Vasile', '5073905481235', '0709326784');
INSERT INTO Informatii_clienti
VALUES(Secventa.nextval, 'NumeTest', 'PrenumeTest', '1111111111111', '0711111111');
INSERT INTO Informatii_clienti
VALUES(Secventa.nextval, 'NumeTest2', 'PrenumeTest2', '1111111111112', '0711111112');
INSERT INTO Informatii_clienti
VALUES(Secventa.nextval, 'NumeTest3', 'PrenumeTest3', '1111111111113', '0711111113');
```

```
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Marian', 'Andrei', '5020413150634', '0734135426');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ion', 'Vasile', '5026262782345', '0774357120');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ioana', 'Cristina', '5022494157062', '0732584214');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Buzatu', 'Ionut', '5021395602456', '0712690369');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ana', 'Cosmina', '5021049258568', '0725790632');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Gigel', 'Frone', '5021256073548', '0724679520');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Andreea', 'Maria', '5021249056732', '0724905687');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Cristi', 'Sebastian', '5042458037893', '0738952469');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Dragos', 'Andrei', '5015730458972', '0739503182');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ionut', 'Dima', '5072490567120', '0719023567');
INSERT INTO Informatii_Clienti
VALUES (Secventa.nextval, 'Ion', 'Matei', '5073905481235', '0709326784');

SELECT *
FROM Informatii_Clienti;
```

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.002 seconds

	COD_INFORMATII_CLIENT	NUME	PRENUME	CNP	NUMAR_TELEFON
1	1	Marian	Andrei	5020413150634	0734135426
2	2	Ion	Vasile	5026262782345	0774357120
3	3	Ioana	Cristina	5022494157062	0732584214
4	4	Buzatu	Ionut	5021395602456	0712690369
5	5	Ana	Cosmina	5021049258568	0725790632
6	6	Gigel	Frone	5021256073548	0724679520
7	7	Andreea	Maria	5021249056732	0724905687
8	8	Cristi	Sebastian	5042458037893	0738952469
9	9	Dragos	Andrei	5015730458972	0739503182
10	10	Ionut	Dima	5072490567120	0719023567
11	11	Ion	Matei	5073905481235	0709326784

- CLIENTI

```
INSERT INTO Clienti
VALUES (Secventa.nextval, 1, 'marian08', 'parolamarian', 'marian08@yahoo.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, NULL, 'john_vasy', '12345678', 'vasy_john@outlook.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, 3, 'Icris', 'crispass', 'criss@gmail.com');
```

```
INSERT INTO Clienti
VALUES (Secventa.nextval, 4, 'buzion', 'parola', 'buzion23@s.unibuc.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, 5, 'anacos', 'cos04322', 'ana@gmail.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, NULL, 'dimaI', 'A45fFD43', 'ion004@yahoo.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 9, 'dragos244', 'parolaMea', 'dragos244@s.unibuc.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, 8, 'seby', 'sebyyy435', 'seb@yahoo.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 7, 'andreeaMaria', '987654321', 'maria_andreea@gmail.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, NULL, 'kingFrone', 'gigel34', 'frone34@outlook.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 2, 'mexicanu', 'meml34', 'mexicanu@outlook.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 11, 'MAtei321', '43mat', 'matei@outlook.com');
INSERT INTO Clienti
VALUES(Secventa.nextval, 12, 'TestUsername' , 'testpass', 'test@yahoo');
INSERT INTO Clienti
VALUES(Secventa.nextval, 13, 'TestUsername2' , 'testpass2', 'test2@yahoo');
INSERT INTO Clienti
VALUES(Secventa.nextval, 14, 'TestUsername3' , 'testpass3', 'test3@yahoo');
```

```
INSERT INTO Clienti
VALUES (Secventa.nextval, 1, 'marian08', 'parolamarian', 'marian08@yahoo.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, NULL, 'john_vasy', '12345678', 'vasy_john@outlook.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, 3, 'Icris', 'crispass', 'criss@gmail.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 4, 'buzion', 'parola', 'buzion23@s.unibuc.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, 5, 'anacos', 'cos04322', 'ana@gmail.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, NULL, 'dimaI', 'A45fFD43', 'ion004@yahoo.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 9, 'dragos244', 'parolaMea', 'dragos244@s.unibuc.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, 8, 'seby', 'sebyyy435', 'seb@yahoo.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 7, 'andreeaMaria', '987654321', 'maria_andreea@gmail.ro');
INSERT INTO Clienti
VALUES (Secventa.nextval, NULL, 'kingFrone', 'gigel34', 'frone34@outlook.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 2, 'mexicanu', 'meml34', 'mexicanu@outlook.com');
INSERT INTO Clienti
VALUES (Secventa.nextval, 11, 'MAtei321', '43mat', 'matei@outlook.com');


SELECT *
FROM Clienti;
```

Script Output | Query Result

SQL | All Rows Fetched: 12 in 0.001 seconds

	COD_CLIENT	COD_INFORMATII_CLIENT	NUME_UTILIZATOR	PAROLA	EMAIL
1	1	1	marian08	parolamarian	marian08@yahoo.com
2	2	(null)	john_vasy	12345678	vasy_john@outlook.ro
3	3	3	Icris	crispass	criss@gmail.com
4	4	4	buzion	parola	buzion23@s.unibuc.ro
5	5	5	anacos	cos04322	ana@gmail.com
6	6	(null)	dimaI	A45fFD43	ion004@yahoo.com
7	7	9	dragos244	parolaMea	dragos244@s.unibuc.ro
8	8	8	seby	sebyyy435	seb@yahoo.com
9	9	7	andreeaMaria	987654321	maria_andreea@gmail.ro
10	10	(null)	kingFrone	gigel34	frone34@outlook.com
11	11	2	mexicanu	meml34	mexicanu@outlook.com
12	12	11	MAtei321	43mat	matei@outlook.com

- INFORMATII_BANCARE

INSERT INTO informatii_bancare
VALUES (1, 1, 200, 150);
INSERT INTO informatii_bancare
VALUES (2, 2, 300, 200);

```
INSERT INTO informatii_bancare (cod_informatii_bancare, cod_client)
VALUES (3, 3);
INSERT INTO informatii_bancare
VALUES (4, 10, 100, 400);
INSERT INTO informatii_bancare
VALUES (5, 9, 500, 270);
INSERT INTO informatii_bancare (cod_informatii_bancare, cod_client)
VALUES (6, 8);
INSERT INTO informatii_bancare
VALUES (7, 7, 850, 1300);
INSERT INTO informatii_bancare
VALUES (8, 6, 2500, 2000);
INSERT INTO informatii_bancare (cod_informatii_bancare, cod_client)
VALUES (9, 5);
INSERT INTO informatii_bancare
VALUES (10, 4, 2700, 4000);
INSERT INTO informatii_bancare
VALUES (11, 12, 2400, 3500);
INSERT INTO Informatii_Bancare
VALUES (12, 13, 3000, 1000);
INSERT INTO Informatii_Bancare
VALUES (13, 14, 2500, 500);
```

```sql
INSERT INTO informatii_bancare
VALUES (1, 1, 200, 150);
INSERT INTO informatii_bancare
VALUES (2, 2, 300, 200);
INSERT INTO informatii_bancare (cod_informatii_bancare, cod_client)
VALUES (3, 3);
INSERT INTO informatii_bancare
VALUES (4, 10, 100, 400);
INSERT INTO informatii_bancare
VALUES (5, 9, 500, 270);
INSERT INTO informatii_bancare (cod_informatii_bancare, cod_client)
VALUES (6, 8);
INSERT INTO informatii_bancare
VALUES (7, 7, 850, 1300);
INSERT INTO informatii_bancare
VALUES (8, 6, 2500, 2000);
INSERT INTO informatii_bancare (cod_informatii_bancare, cod_client)
VALUES (9, 5);
INSERT INTO informatii_bancare
VALUES (10, 4, 2700, 4000);
INSERT INTO informatii_bancare
VALUES (11, 12, 2400, 3500);

SELECT *
FROM Informatii_Bancare;
```

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.002 seconds

	COD_INFORMATII_BANCARE	COD_CLIENT	SOLD_CURENT	SUMA_CHELTUITA
1	1	1	200	150
2	2	2	300	200
3	3	3	0	0
4	4	10	100	400
5	5	9	500	270
6	6	8	0	0
7	7	7	850	1300
8	8	6	2500	2000
9	9	5	0	0
10	10	4	2700	4000
11	11	12	2400	3500

- CARDURI_BANCARE

INSERT INTO Carduri_Bancare

```
VALUES (1, 1, 1, '4053123405931285', TO_DATE('23-MAR-2027', 'DD-MON-YYYY'),
643);
INSERT INTO Carduri_Bancare
VALUES (2, 2, 2, '5902345967102923', TO_DATE('03-APR-2028', 'DD-MON-YYYY'),
346);
INSERT INTO Carduri_Bancare
VALUES (3, 3, 3, '5678765321426541', TO_DATE('14-DEC-2025', 'DD-MON-YYYY'),
964);
INSERT INTO Carduri_Bancare
VALUES (4, 4, 10, '876541234765423', TO_DATE('18-FEB-2024', 'DD-MON-YYYY'),
245);
INSERT INTO Carduri_Bancare
VALUES (5, 5, 9, '2134357568505663', TO_DATE('22-AUG-2029', 'DD-MON-YYYY'),
325);
INSERT INTO Carduri_Bancare
VALUES (6, 6, 8, '7809675421345633', TO_DATE('27-SEP-2025', 'DD-MON-YYYY'),
234);
INSERT INTO Carduri_Bancare
VALUES (7, 7, 7, '1242365406784353', TO_DATE('22-DEC-2025', 'DD-MON-YYYY'),
641);
INSERT INTO Carduri_Bancare
VALUES (8, 8, 6, '1234326347314746', TO_DATE('06-OCT-2030', 'DD-MON-YYYY'),
123);
INSERT INTO Carduri_Bancare
VALUES (9, 9, 5, '7437217457897523', TO_DATE('08-NOV-2023', 'DD-MON-YYYY'),
435);
INSERT INTO Carduri_Bancare
VALUES (10, 10, 4, '3242356343473442', TO_DATE('17-APR-2028', 'DD-MON-YYYY'),
543);
INSERT INTO Carduri_Bancare
VALUES (11, 11, 12, '3202356343473569', TO_DATE('26-DEC-2030', 'DD-MON-YYYY'),
206);
INSERT INTO Carduri_Bancare
VALUES (12, 12, 13, '1111111111111111', TO_DATE('12-DEC-2050', 'DD-MON-YYYY'),
111);
INSERT INTO Carduri_Bancare
VALUES (13, 12, 13, '2222222222222222', TO_DATE('12-DEC-2070', 'DD-MON-YYYY'),
222);
INSERT INTO Carduri_Bancare
VALUES (14, 12, 13, '3333333333333333', TO_DATE('12-DEC-2090', 'DD-MON-YYYY'),
222);
INSERT INTO Carduri_Bancare
VALUES (15, 13, 14, '1111111111111122', TO_DATE('12-DEC-2050', 'DD-MON-YYYY'),
111);
```

INSERT INTO Carduri_Bancare
VALUES (16, 13, 14, '2222222222222233', TO_DATE('12-DEC-2070', 'DD-MON-YYYY'), 222);

```
INSERT INTO Carduri_Bancare
VALUES (1, 1, 1, '4053123405931285', TO_DATE('23-MAR-2027', 'DD-MON-YYYY'), 643);
INSERT INTO Carduri_Bancare
VALUES (2, 2, 2, '5902345967102923', TO_DATE('03-APR-2028', 'DD-MON-YYYY'), 346);
INSERT INTO Carduri_Bancare
VALUES (3, 3, 3, '5678765321426541', TO_DATE('14-DEC-2025', 'DD-MON-YYYY'), 964);
INSERT INTO Carduri_Bancare
VALUES (4, 4, 10, '876541234765423', TO_DATE('18-FEB-2024', 'DD-MON-YYYY'), 245);
INSERT INTO Carduri_Bancare
VALUES (5, 5, 9, '2134357568505663', TO_DATE('22-AUG-2029', 'DD-MON-YYYY'), 325);
INSERT INTO Carduri_Bancare
VALUES (6, 6, 8, '7809675421345633', TO_DATE('27-SEP-2025', 'DD-MON-YYYY'), 234);
INSERT INTO Carduri_Bancare
VALUES (7, 7, 7, '1242365406784353', TO_DATE('22-DEC-2025', 'DD-MON-YYYY'), 641);
INSERT INTO Carduri_Bancare
VALUES (8, 8, 6, '1234326347314746', TO_DATE('06-OCT-2030', 'DD-MON-YYYY'), 123);
INSERT INTO Carduri_Bancare
VALUES (9, 9, 5, '7437217457897523', TO_DATE('08-NOV-2023', 'DD-MON-YYYY'), 435);
INSERT INTO Carduri_Bancare
VALUES (10, 10, 4, '3242356343473442', TO_DATE('17-APR-2028', 'DD-MON-YYYY'), 543);
INSERT INTO Carduri_Bancare
VALUES (11, 11, 12, '3202356343473569', TO_DATE('26-DEC-2030', 'DD-MON-YYYY'), 206);


SELECT *
FROM Carduri_Bancare;
```

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.002 seconds

	COD_CARD_BANCAR	COD_INFORMATII_BANCARE	COD_CLIENT	NUMAR_CARD	DATA_EXPIRARE_CARD	COD_SECURITATE_CARD
1	1	1	1	4053123405931285	23-MAR-27	643
2	2	2	2	5902345967102923	03-APR-28	346
3	3	3	3	5678765321426541	14-DEC-25	964
4	4	4	10	876541234765423	18-FEB-24	245
5	5	5	9	2134357568505663	22-AUG-29	325
6	6	6	8	7809675421345633	27-SEP-25	234
7	7	7	7	1242365406784353	22-DEC-25	641
8	8	8	6	1234326347314746	06-OCT-30	123
9	9	9	5	7437217457897523	08-NOV-23	435
10	10	10	4	3242356343473442	17-APR-28	543
11	11	11	12	3202356343473569	26-DEC-30	206

- ADRESE

INSERT INTO Adrese
VALUES (1, 1, 'Romania', 'Craiova', 'Brestei', 56);
INSERT INTO Adrese
VALUES (2, 1, 'Romania', 'Bucuresti', 'Drumul Taberei', 23);
INSERT INTO Adrese
VALUES (3, 1, 'Olanda', 'Amsterdam', 'Amsterdam Street', 4535);
INSERT INTO Adrese
VALUES (1, 4, 'Romania', 'Iasi', 'Strada Iasului', 654);

```
INSERT INTO Adrese
VALUES (2, 4, 'Romania', 'Timisioara', 'Aleea Actorilor', 547);
INSERT INTO Adrese
VALUES (1, 7, 'Republica Moldova', 'Chisinau', 'Strada Mosilor', 234);
INSERT INTO Adrese
VALUES (1, 8, 'Franta', 'Paris', 'Sans Elise', 1235);
INSERT INTO Adrese
VALUES (1, 10, 'Anglia', 'Londra', 'Fournier Street', 12);
INSERT INTO Adrese
VALUES (2, 10, 'Romania', 'Constanta', 'Faleza Marii', 634);
INSERT INTO Adrese
VALUES (3, 10, 'Romania', 'Cluj', 'Strada Clujului', 8765);
INSERT INTO Adrese
VALUES (1, 2, 'Romania', 'Arad', 'Strda Aradului', 65);
INSERT INTO Adrese
VALUES (2, 2, 'Romania', 'Bucuresti', 'Crangasi', 654);
INSERT INTO Adrese
VALUES (1, 9, 'Romania', 'Buzau', 'Strda Buzaului', 23);
INSERT INTO Adrese
VALUES (1, 11, 'Romania', 'Botosani', 'Strda Botosani', 653);
INSERT INTO Adrese
VALUES (1, 12, 'Romania', 'Vaslui', 'Strda Vslui', 13);
INSERT INTO Adrese
VALUES (1, 13, 'Romania', 'Craiova', 'Stadionului', 80);
INSERT INTO Adrese VALUES
(1, 14, 'Romania', 'Craiova', 'Stadionului', 80);
INSERT INTO Adrese
VALUES (1, 15, 'Romania', 'Craiova', 'Stadionului', 80);
```

Worksheet | Query Builder

```
VALUES (2, 1, 'Romania', 'Bucuresti', 'Drumul Taberei', 23);
INSERT INTO Adrese
VALUES (3, 1, 'Olanda', 'Amsterdam', 'Amsterdam Street', 4535);
INSERT INTO Adrese
VALUES (1, 4, 'Romania', 'Iasi', 'Strada Iasului', 654);
INSERT INTO Adrese
VALUES (2, 4, 'Romania', 'Timisioara', 'Aleea Actorilor', 547);
INSERT INTO Adrese
VALUES (1, 7, 'Republica Moldova', 'Chisinau', 'Strada Mosilor', 234);
INSERT INTO Adrese
VALUES (1, 8, 'Franta', 'Paris', 'Sans Elise', 1235);
INSERT INTO Adrese
VALUES (1, 10, 'Anglia', 'Londra', 'Fournier Street', 12);
INSERT INTO Adrese
VALUES (2, 10, 'Romania', 'Constanta', 'Faleza Marii', 634);
INSERT INTO Adrese
VALUES (3, 10, 'Romania', 'Cluj', 'Strada Clujului', 8765);
INSERT INTO Adrese
VALUES (1, 2, 'Romania', 'Arad', 'Strda Aradului', 65);
INSERT INTO Adrese
VALUES (2, 2, 'Romania', 'Bucuresti', 'Crangasi', 654);
INSERT INTO Adrese
VALUES (1, 9, 'Romania', 'Buzau', 'Strda Buzaului', 23);
INSERT INTO Adrese
VALUES (1, 11, 'Romania', 'Botosani', 'Strda Botosani', 653);
INSERT INTO Adrese
VALUES (1, 12, 'Romania', 'Vaslui', 'Strda Vslui', 13);

SELECT *
FROM Adrese;
```

Script Output | Query Result

SQL | All Rows Fetched: 15 in 0.002 seconds

	COD_ADRESA	COD_CLIENT	TARA	ORAS	STRADA	NUMAR
4	1	4	Romania	Iasi	Strada Iasului	654
5	2	4	Romania	Timisioara	Aleea Actorilor	547
6	1	7	Republica Moldova	Chisinau	Strada Mosilor	234
7	1	8	Franta	Paris	Sans Elise	1235
8	1	10	Anglia	Londra	Fournier Street	12
9	2	10	Romania	Constanta	Faleza Marii	634
10	3	10	Romania	Cluj	Strada Clujului	8765
11	1	2	Romania	Arad	Strda Aradului	65
12	2	2	Romania	Bucuresti	Crangasi	654
13	1	9	Romania	Buzau	Strda Buzaului	23
14	1	11	Romania	Botosani	Strda Botosani	653
15	1	12	Romania	Vaslui	Strda Vslui	13

- FACTURI

INSERT INTO Facturi

```
VALUES (1, 1, 1, 200, TO_DATE('18-FEB-2024', 'DD-MON-YYYY'),
TO_DATE('23-FEB-2025', 'DD-MON-YYYY'), 'Platit');
INSERT INTO Facturi
VALUES (2, 1, 1, 350, TO_DATE('04-MAR-2023', 'DD-MON-YYYY'),
TO_DATE('15-APR-2024', 'DD-MON-YYYY'), 'In procesare');
INSERT INTO Facturi
VALUES (3, 1, 1, 570, TO_DATE('20-SEP-2022', 'DD-MON-YYYY'),
TO_DATE('01-APR-2026', 'DD-MON-YYYY'), 'Platit');
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare,
termen_plata)
VALUES (1, 2, 4, 700, TO_DATE('13-DEC-2022', 'DD-MON-YYYY'),
TO_DATE('17-APR-2024', 'DD-MON-YYYY'));
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare,
termen_plata)
VALUES (2, 2, 4, 620, TO_DATE('07-AUG-2024', 'DD-MON-YYYY'),
TO_DATE('15-APR-2028', 'DD-MON-YYYY'));
INSERT INTO Facturi
VALUES (1, 1, 7, 200, TO_DATE('23-FEB-2023', 'DD-MON-YYYY'),
TO_DATE('27-OCT-2027', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (2, 1, 7, 850, TO_DATE('20-AUG-2023', 'DD-MON-YYYY'),
TO_DATE('27-APR-2025', 'DD-MON-YYYY'), 'In procesare');
INSERT INTO Facturi
VALUES (1, 2, 10,340, TO_DATE('14-JAN-2023', 'DD-MON-YYYY'),
TO_DATE('19-NOV-2023', 'DD-MON-YYYY'), 'Platit');
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare,
termen_plata)
VALUES (2, 2, 10, 890, TO_DATE('27-DEC-2024', 'DD-MON-YYYY'),
TO_DATE('08-APR-2027', 'DD-MON-YYYY'));
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare,
termen_plata)
VALUES (3, 2, 10, 510, TO_DATE('03-APR-2023', 'DD-MON-YYYY'),
TO_DATE('22-APR-2025', 'DD-MON-YYYY'));
INSERT INTO Facturi
VALUES (1, 1, 2, 280, TO_DATE('24-SEP-2020', 'DD-MON-YYYY'),
TO_DATE('27-APR-2022', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 9, 320, TO_DATE('14-DEC-2022', 'DD-MON-YYYY'),
TO_DATE('25-MAR-2025', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 11, 765, TO_DATE('05-FEB-2022', 'DD-MON-YYYY'),
TO_DATE('02-MAY-2024', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
```

```
VALUES (1, 1, 12, 180, TO_DATE('19-APR-2021', 'DD-MON-YYYY'),
TO_DATE('02-APR-2022', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 13, 2000, TO_DATE('12-DEC-2020', 'DD-MON-YYYY'),
TO_DATE('15-FEB-2025', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi VALUES
 (1, 1, 14, 2000, TO_DATE('12-DEC-2020', 'DD-MON-YYYY'), TO_DATE('15-FEB-2025',
'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi VALUES
 (2, 1, 14, 3000, TO_DATE('01-APR-2019', 'DD-MON-YYYY'), TO_DATE('23-DEC-2024',
'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi VALUES
 (3, 1, 14, 1700, TO_DATE('12-DEC-2020', 'DD-MON-YYYY'), TO_DATE('15-FEB-2025',
'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi VALUES
 (4, 1, 14, 2700, TO_DATE('01-APR-2019', 'DD-MON-YYYY'), TO_DATE('23-DEC-2024',
'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 15, 3000, TO_DATE('12-DEC-2020', 'DD-MON-YYYY'),
TO_DATE('15-FEB-2025', 'DD-MON-YYYY'), 'Neplatit');
```

Worksheet | Query Builder

```
VALUES (2, 1, 1, 350, TO_DATE('04-MAR-2023', 'DD-MON-YYYY'), TO_DATE('15-APR-2024', 'DD-MON-YYYY'), 'In procesare');
INSERT INTO Facturi
VALUES (3, 1, 1, 570, TO_DATE('20-SEP-2022', 'DD-MON-YYYY'), TO_DATE('01-APR-2026', 'DD-MON-YYYY'), 'Platit');
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare, termen_plata)
VALUES (1, 2, 4, 700, TO_DATE('13-DEC-2022', 'DD-MON-YYYY'), TO_DATE('17-APR-2024', 'DD-MON-YYYY'));
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare, termen_plata)
VALUES (2, 2, 4, 620, TO_DATE('07-AUG-2024', 'DD-MON-YYYY'), TO_DATE('15-APR-2028', 'DD-MON-YYYY'));
INSERT INTO Facturi
VALUES (1, 1, 7, 200, TO_DATE('23-FEB-2023', 'DD-MON-YYYY'), TO_DATE('27-OCT-2027', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (2, 1, 7, 850, TO_DATE('20-AUG-2023', 'DD-MON-YYYY'), TO_DATE('27-APR-2025', 'DD-MON-YYYY'), 'In procesare');
INSERT INTO Facturi
VALUES (1, 2, 10,340, TO_DATE('14-JAN-2023', 'DD-MON-YYYY'), TO_DATE('19-NOV-2023', 'DD-MON-YYYY'), 'Platit');
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare, termen_plata)
VALUES (2, 2, 10, 890, TO_DATE('27-DEC-2024', 'DD-MON-YYYY'), TO_DATE('08-APR-2027', 'DD-MON-YYYY'));
INSERT INTO Facturi (cod_factura, cod_adresa, cod_client, total, data_eliberare, termen_plata)
VALUES (3, 2, 10, 510, TO_DATE('03-APR-2023', 'DD-MON-YYYY'), TO_DATE('22-APR-2025', 'DD-MON-YYYY'));
INSERT INTO Facturi
VALUES (1, 1, 2, 280, TO_DATE('24-SEP-2020', 'DD-MON-YYYY'), TO_DATE('27-APR-2022', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 9, 320, TO_DATE('14-DEC-2022', 'DD-MON-YYYY'), TO_DATE('25-MAR-2025', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 11, 765, TO_DATE('05-FEB-2022', 'DD-MON-YYYY'), TO_DATE('02-MAY-2024', 'DD-MON-YYYY'), 'Neplatit');
INSERT INTO Facturi
VALUES (1, 1, 12, 180, TO_DATE('19-APR-2021', 'DD-MON-YYYY'), TO_DATE('02-APR-2022', 'DD-MON-YYYY'), 'Neplatit');


SELECT *
FROM Facturi;
```

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.003 seconds

	COD_FACTURA	COD_ADRESA	COD_CLIENT	TOTAL	DATA_ELIBERARE	TERMEN_PLATA	STATUS
3	3	1	1	570	20-SEP-22	01-APR-26	Platit
4	1	2	4	700	13-DEC-22	17-APR-24	Neplatit
5	2	2	4	620	07-AUG-24	15-APR-28	Neplatit
6	1	1	7	200	23-FEB-23	27-OCT-27	Neplatit
7	2	1	7	850	20-AUG-23	27-APR-25	In procesare
8	1	2	10	340	14-JAN-23	19-NOV-23	Platit
9	2	2	10	890	27-DEC-24	08-APR-27	Neplatit
10	3	2	10	510	03-APR-23	22-APR-25	Neplatit
11	1	1	2	280	24-SEP-20	27-APR-22	Neplatit
12	1	1	9	320	14-DEC-22	25-MAR-25	Neplatit
13	1	1	11	765	05-FEB-22	02-MAY-24	Neplatit
14	1	1	12	180	19-APR-21	02-APR-22	Neplatit

- PLANGERI

INSERT INTO Plangeri
VALUES (1, 1, TO_DATE('03-APR-2023', 'DD-MON-YYYY'), 'Romania, Bucuresti, Strada Giulesti, Nr 20', 'S-a spart o teava de gaz');
INSERT INTO Plangeri
VALUES (2, 1, TO_DATE('12-AUG-2023', 'DD-MON-YYYY'), 'Romania, Craiova, Strada Brestei, Nr 35', 'Curge apa din perete');
INSERT INTO Plangeri
VALUES (1, 3, TO_DATE('22-SEP-2020', 'DD-MON-YYYY'), 'Olanda, Amsterdam, Strada Lalelelor, Nr 44', 'S-a luat curentul');
INSERT INTO Plangeri (cod_plangere, cod_client, adresa, mesaj)

VALUES (2, 3, 'Romania, Cluj, Strada Florilor, Nr 120', 'A sarit o siguranta');
INSERT INTO Plangeri
VALUES (1, 4, TO_DATE('02-JUN-2020', 'DD-MON-YYYY'), 'Romania, Bucuresti, Strada Unirii, Nr 67', 'Nu este lumina pe strada');
INSERT INTO Plangeri
VALUES (1, 6, TO_DATE('14-FEB-2022', 'DD-MON-YYYY'), 'Romania, Timisoara, Strada Palatului, Nr 123', 'Nu functioneaza internetul');
INSERT INTO Plangeri
VALUES (2, 6, TO_DATE('17-DEC-2021', 'DD-MON-YYYY'), 'Romania, Oradea, Strada Moldovei, Nr 45', 'Nu avem apa calda');
INSERT INTO Plangeri (cod_plangere, cod_client, adresa, mesaj)
VALUES (1, 9, 'Romania, Craiova, Strada Craiovita, Nr 56', 'S-a ars siguranta pe bloc');
INSERT INTO Plangeri (cod_plangere, cod_client, adresa, mesaj)
VALUES (2, 9, 'Romania, Oradea, Strada Raului, Nr 12', 'S-a luat apa de 3 zile');
INSERT INTO Plangeri
VALUES (1, 10, TO_DATE('23-NOV-2020', 'DD-MON-YYYY'), 'Romania, Oradea, Strada Marii, Nr 67', 'Nu avem apa');

```
INSERT INTO Plangeri
VALUES (1, 1, TO_DATE('03-APR-2023', 'DD-MON-YYYY'), 'Romania, Bucuresti, Strada Giulesti, Nr 20', 'S-a spart o teava de gaz');
INSERT INTO Plangeri
VALUES (2, 1, TO_DATE('12-AUG-2023', 'DD-MON-YYYY'), 'Romania, Craiova, Strada Brestei, Nr 35', 'Curge apa din perete');
INSERT INTO Plangeri
VALUES (1, 3, TO_DATE('22-SEP-2020', 'DD-MON-YYYY'), 'Olanda, Amsterdam, Strada Lalelelor, Nr 44', 'S-a luat curentul');
INSERT INTO Plangeri (cod_plangere, cod_client, adresa, mesaj)
VALUES (2, 3, 'Romania, Cluj, Strada Florilor, Nr 120', 'A sarit o siguranta');
INSERT INTO Plangeri
VALUES (1, 4, TO_DATE('02-JUN-2020', 'DD-MON-YYYY'), 'Romania, Bucuresti, Strada Unirii, Nr 67', 'Nu este lumina pe strada');
INSERT INTO Plangeri
VALUES (1, 6, TO_DATE('14-FEB-2022', 'DD-MON-YYYY'), 'Romania, Timisoara, Strada Palatului, Nr 123', 'Nu functioneaza internetul');
INSERT INTO Plangeri
VALUES (2, 6, TO_DATE('17-DEC-2021', 'DD-MON-YYYY'), 'Romania, Oradea, Strada Moldovei, Nr 45', 'Nu avem apa calda');
INSERT INTO Plangeri (cod_plangere, cod_client, adresa, mesaj)
VALUES (1, 9, 'Romania, Craiova, Strada Craiovita, Nr 56', 'S-a ars siguranta pe bloc');
INSERT INTO Plangeri (cod_plangere, cod_client, adresa, mesaj)
VALUES (2, 9, 'Romania, Oradea, Strada Raului, Nr 12', 'S-a luat apa de 3 zile');
INSERT INTO Plangeri
VALUES (1, 10, TO_DATE('23-NOV-2020', 'DD-MON-YYYY'), 'Romania, Oradea, Strada Marii, Nr 67', 'Nu avem apa');


SELECT *
FROM Plangeri;
```

Script Output × Query Result ×

SQL | All Rows Fetched: 10 in 0.002 seconds

	COD_PLANGERE	COD_CLIENT	DATA_PLANGERE	ADRESA	MESAJ
1	1	1	03-APR-23	Romania, Bucuresti, Strada Giulesti, Nr 20	S-a spart o teava de gaz
2	2	1	12-AUG-23	Romania, Craiova, Strada Brestei, Nr 35	Curge apa din perete
3	1	3	22-SEP-20	Olanda, Amsterdam, Strada Lalelelor, Nr 44	S-a luat curentul
4	2	3	15-MAY-22	Romania, Cluj, Strada Florilor, Nr 120	A sarit o siguranta
5	1	4	02-JUN-20	Romania, Bucuresti, Strada Unirii, Nr 67	Nu este lumina pe strada
6	1	6	14-FEB-22	Romania, Timisoara, Strada Palatului, Nr 123	Nu functioneaza internetul
7	2	6	17-DEC-21	Romania, Oradea, Strada Moldovei, Nr 45	Nu avem apa calda
8	1	9	15-MAY-22	Romania, Craiova, Strada Craiovita, Nr 56	S-a ars siguranta pe bloc
9	2	9	15-MAY-22	Romania, Oradea, Strada Raului, Nr 12	S-a luat apa de 3 zile
10	1	10	23-NOV-20	Romania, Oradea, Strada Marii, Nr 67	Nu avem apa

- INFORMATII_MUNCITORI

```
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ion', 'Andrei', '5021256073548', '0724679520');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Mincu', 'Ionut', '5023490567123', '0730941278');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ioana', 'Maria', '5034091285674', '0710923856');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Cosmin', 'Marian', '5021092375834', '0730923857');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Loredana', 'Mariana', '5039054673120', '0745093285');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Sebastian', 'Marius', '5021093457835', '0749012356');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Oana', 'Floarea', '5038902135672', '0709431846');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ion', 'Antonescu', '5024095689123', '0756908323');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Marius', 'Lica', '5021905623895', '0701938567');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Sorin', 'Mircea', '5023095671834', '0789042175');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Miruna', 'Mircea', '5023095671309', '0789042111');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ionut', 'Ionel', '5010495671309', '0709342111');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Vasile', 'Vasilievici', '5023095673091', '0789043331');
```

Worksheet | Query Builder

```
VALUES (SECVENTA.nextval, 'Ion', 'Andrei', '5021256073548', '0724679520');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Mincu', 'Ionut', '5023490567123', '0730941278');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ioana', 'Maria', '5034091285674', '0710923856');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Cosmin', 'Marian', '5021092375834', '0730923857');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Loredana', 'Mariana', '5039054673120', '0745093285');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Sebastian', 'Marius', '5021093457835', '0749012356');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Oana', 'Floarea', '5038902135672', '0709431846');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ion', 'Antonescu', '5024095689123', '0756908323');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Marius', 'Lica', '5021905623895', '0701938567');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Sorin', 'Mircea', '5023095671834', '0789042175');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Miruna', 'Mircea', '5023095671309', '0789042111');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Ionut', 'Ionel', '5010495671309', '0709342111');
INSERT INTO Informatii_Muncitori
VALUES (SECVENTA.nextval, 'Vasile', 'Vasilievici', '5023095673091', '0789043331');

SELECT *
FROM Informatii_Muncitori;
```

Script Output | Query Result

SQL | All Rows Fetched: 13 in 0.002 seconds

	COD_INFORMATII_MUNCITOR	NUME	PRENUME	CNP	NUMAR_TELEFON
2	2	Mincu	Ionut	5023490567123	0730941278
3	3	Ioana	Maria	5034091285674	0710923856
4	4	Cosmin	Marian	5021092375834	0730923857
5	5	Loredana	Mariana	5039054673120	0745093285
6	6	Sebastian	Marius	5021093457835	0749012356
7	7	Oana	Floarea	5038902135672	0709431846
8	8	Ion	Antonescu	5024095689123	0756908323
9	9	Marius	Lica	5021905623895	0701938567
10	10	Sorin	Mircea	5023095671834	0789042175
11	11	Miruna	Mircea	5023095671309	0789042111
12	12	Ionut	Ionel	5010495671309	0709342111
13	13	Vasile	Vasilievici	5023095673091	0789043331

- MUNCITORI

INSERT INTO Muncitori
VALUES (1, 1, 2300, TO_DATE('23-NOV-2019', 'DD-MON-YYYY'), 20);
INSERT INTO Muncitori

```
VALUES (2, 2, 1800, TO_DATE('10-SEP-2020', 'DD-MON-YYYY'), 50);
INSERT INTO Muncitori
VALUES (3, NULL, 1000, TO_DATE('27-DEC-2021', 'DD-MON-YYYY'), 10);
INSERT INTO Muncitori
VALUES (4, 6, 3000, TO_DATE('15-JAN-2018', 'DD-MON-YYYY'), 200);
INSERT INTO Muncitori
VALUES (5, 8, 8000, TO_DATE('01-OCT-2012', 'DD-MON-YYYY'), 800);
INSERT INTO Muncitori
VALUES (6, 5, 4000, TO_DATE('14-AUG-2017', 'DD-MON-YYYY'), 400);
INSERT INTO Muncitori
VALUES (7, NULL, 2300, TO_DATE('20-FEB-2020', 'DD-MON-YYYY'), NULL);
INSERT INTO Muncitori
VALUES (8, 7, 3300, TO_DATE('19-JUL-2020', 'DD-MON-YYYY'), 70);
INSERT INTO Muncitori
VALUES (9, 9, 1700, TO_DATE('08-MAR-2021', 'DD-MON-YYYY'), 30);
INSERT INTO Muncitori
VALUES (10, 3, 1600, TO_DATE('25-MAY-2017', 'DD-MON-YYYY'), NULL);
INSERT INTO Muncitori
VALUES (11, 11, 1200, TO_DATE('13-DEC-2020', 'DD-MON-YYYY'), 456);
INSERT INTO Muncitori
VALUES (12, 12, 2000, TO_DATE('24-SEP-2020', 'DD-MON-YYYY'), 555);
INSERT INTO Muncitori
VALUES (13, 13, 1300, TO_DATE('13-DEC-2017', 'DD-MON-YYYY'), 456);
```

```sql
INSERT INTO Muncitori
VALUES (1, 1, 2300, TO_DATE('23-NOV-2019', 'DD-MON-YYYY'), 20);
INSERT INTO Muncitori
VALUES (2, 2, 1800, TO_DATE('10-SEP-2020', 'DD-MON-YYYY'), 50);
INSERT INTO Muncitori
VALUES (3, NULL, 1000, TO_DATE('27-DEC-2021', 'DD-MON-YYYY'), 10);
INSERT INTO Muncitori
VALUES (4, 6, 3000, TO_DATE('15-JAN-2018', 'DD-MON-YYYY'), 200);
INSERT INTO Muncitori
VALUES (5, 8, 8000, TO_DATE('01-OCT-2012', 'DD-MON-YYYY'), 800);
INSERT INTO Muncitori
VALUES (6, 5, 4000, TO_DATE('14-AUG-2017', 'DD-MON-YYYY'), 400);
INSERT INTO Muncitori
VALUES (7, NULL, 2300, TO_DATE('20-FEB-2020', 'DD-MON-YYYY'), NULL);
INSERT INTO Muncitori
VALUES (8, 7, 3300, TO_DATE('19-JUL-2020', 'DD-MON-YYYY'), 70);
INSERT INTO Muncitori
VALUES (9, 9, 1700, TO_DATE('08-MAR-2021', 'DD-MON-YYYY'), 30);
INSERT INTO Muncitori
VALUES (10, 3, 1600, TO_DATE('25-MAY-2017', 'DD-MON-YYYY'), NULL);
INSERT INTO Muncitori
VALUES (11, 11, 1200, TO_DATE('13-DEC-2020', 'DD-MON-YYYY'), 456);
INSERT INTO Muncitori
VALUES (12, 12, 2000, TO_DATE('24-SEP-2020', 'DD-MON-YYYY'), 555);
INSERT INTO Muncitori
VALUES (13, 13, 1300, TO_DATE('13-DEC-2017', 'DD-MON-YYYY'), 456);

SELECT *
FROM Muncitori;
```

Script Output | Query Result

SQL | All Rows Fetched: 13 in 0.002 seconds

	COD_MUNCITOR	COD_INFORMATII_MUNCITOR	SALARIU	DATA_ANGAJARE	RATING
2	2	2	1800	10-SEP-20	50
3	3	(null)	1000	27-DEC-21	10
4	4	6	3000	15-JAN-18	200
5	5	8	8000	01-OCT-12	800
6	6	5	4000	14-AUG-17	400
7	7	(null)	2300	20-FEB-20	(null)
8	8	7	3300	19-JUL-20	70
9	9	9	1700	08-MAR-21	30
10	10	3	1600	25-MAY-17	(null)
11	11	11	1200	13-DEC-20	456
12	12	12	2000	24-SEP-20	555
13	13	13	1300	13-DEC-17	456

- SPECIALIZARI

```
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Sudor', 'Sudeaza tevile sparte');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Sofer', 'Conduce masina pana la locul interventiei');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Instalator', 'Se ocupa cu montatul diverselor instalatii');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Supervizor', 'Se ocupa cu coordonarea interventiilor');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Verificator gaze', 'Verifica tevile si instalatiile de gaze');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Stivuitorist', 'Manevreaza stivuitoare');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Electrician', 'Repara si inspecteaza instalatiile electrice');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Retelist', 'Se ocupa cu intretinerea echipamentelor de internet');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Tinichigiu', 'Repara si indreapta tevile de apa sparte si fisurate');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Buldozerist', 'Manevreaza buldozere');
```

```
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Sudor', 'Sudeaza tevile sparte');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Sofer', 'Conduce masina pana la locul interventiei');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Instalator', 'Se ocupa cu montatul diverselor instalatii');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Supervizor', 'Se ocupa cu coordonarea interventiilor');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Verificator gaze', 'Verifica tevile si instalatiile de gaze');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Stivuitorist', 'Manevreaza stivuitoare');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Electrician', 'Repara si inspecteaza instalatiile electrice');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Retelist', 'Se ocupa cu intretinerea echipamentelor de internet');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Tinichigiu', 'Repara si indreapta tevile de apa sparte si fisurate');
INSERT INTO Specializari
VALUES (Secventa.nextval, 'Buldozerist', 'Manevreaza buldozere');


SELECT *
FROM Specializari;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.001 seconds

	COD_SPECIALIZARE	NUME_SPECIALIZARE	DESCRIERE
1	2	Sofer	Conduce masina pana la locul interventiei
2	3	Instalator	Se ocupa cu montatul diverselor instalatii
3	4	Supervizor	Se ocupa cu coordonarea interventiilor
4	5	Verificator gaze	Verifica tevile si instalatiile de gaze
5	6	Stivuitorist	Manevreaza stivuitoare
6	7	Electrician	Repara si inspecteaza instalatiile electrice
7	8	Retelist	Se ocupa cu intretinerea echipamentelor de internet
8	9	Tinichigiu	Repara si indreapta tevile de apa sparte si fisurate
9	10	Buldozerist	Manevreaza buldozere
10	1	Sudor	Sudeaza tevile sparte

- SPECIALIZARI_MUNCITORI

INSERT INTO Specializari_Muncitori
VALUES (1, 1);
INSERT INTO Specializari_Muncitori
VALUES (1, 9);
INSERT INTO Specializari_Muncitori
VALUES (2, 2);
INSERT INTO Specializari_Muncitori
VALUES (3, 6);
INSERT INTO Specializari_Muncitori
VALUES (3, 10);
INSERT INTO Specializari_Muncitori
VALUES (4, 8);
INSERT INTO Specializari_Muncitori

```
VALUES (5, 4);
INSERT INTO Specializari_Muncitori
VALUES (6, 3);
INSERT INTO Specializari_Muncitori
VALUES (6, 5);
INSERT INTO Specializari_Muncitori
VALUES (6, 7);
INSERT INTO Specializari_Muncitori
VALUES (8, 2);
INSERT INTO Specializari_Muncitori
VALUES (9, 5);
INSERT INTO Specializari_Muncitori
VALUES (10, 2);
INSERT INTO Specializari_Muncitori
VALUES (10, 8);
INSERT INTO Specializari_Muncitori
VALUES (1, 2);
INSERT INTO Specializari_Muncitori
VALUES (11, 2);
INSERT INTO Specializari_Muncitori
VALUES (12, 2);
INSERT INTO Specializari_Muncitori
VALUES (13, 2);
INSERT INTO Specializari_Muncitori
VALUES (1, 7);
INSERT INTO Specializari_Muncitori
VALUES (1, 8);
INSERT INTO Specializari_Muncitori
VALUES (10, 1);
```

Worksheet | Query Builder

```
VALUES (6, 5);
INSERT INTO Specializari_Muncitori
VALUES (6, 7);
INSERT INTO Specializari_Muncitori
VALUES (8, 2);
INSERT INTO Specializari_Muncitori
VALUES (9, 5);
INSERT INTO Specializari_Muncitori
VALUES (10, 2);
INSERT INTO Specializari_Muncitori
VALUES (10, 8);
INSERT INTO Specializari_Muncitori
VALUES (1, 2);
INSERT INTO Specializari_Muncitori
VALUES (11, 2);
INSERT INTO Specializari_Muncitori
VALUES (12, 2);
INSERT INTO Specializari_Muncitori
VALUES (13, 2);
INSERT INTO Specializari_Muncitori
VALUES (1, 7);
INSERT INTO Specializari_Muncitori
VALUES (1, 8);
INSERT INTO Specializari_Muncitori
VALUES (10, 1);

SELECT *
FROM Specializari_Muncitori;
```

Script Output | Query Result

SQL | All Rows Fetched: 21 in 0.001 seconds

	COD_MUNCITOR	COD_SPECIALIZARE
10	5	4
11	6	3
12	6	5
13	6	7
14	8	2
15	9	5
16	10	1
17	10	2
18	10	8
19	11	2
20	12	2
21	13	2

- INTERVENTII

```
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('01-OCT-2021', 'DD-MON-YYYY'), 'Teava de apa sparta', 'In lucuru');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('12-DEC-2022', 'DD-MON-YYYY'), 'Scurgere de gaze', 'Neinceputa');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('25-NOV-2025', 'DD-MON-YYYY'), 'Teava de gaze sparte', 'Finalizata');
INSERT INTO Interventii (cod_interventie, data_incepere, nume_defectiune)
VALUES (Secventa.nextval, TO_DATE('15-MAY-2022', 'DD-MON-YYYY'), 'Cablu de electricitate intrerupt');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('20-SEP-2021', 'DD-MON-YYYY'), 'Internet picat', 'Finalizata');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('05-MAY-2022', 'DD-MON-YYYY'), 'Apa nefunctionala', 'In lucru');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('19-APR-2021', 'DD-MON-YYYY'), 'Siguranta sarita', 'In lucru');
INSERT INTO Interventii (cod_interventie, data_incepere, nume_defectiune)
VALUES (Secventa.nextval, TO_DATE('20-JUN-2025', 'DD-MON-YYYY'), 'Teava de apa fisurata');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('23-APR-2023', 'DD-MON-YYYY'), 'Siguranta sarita', 'Neinceputa');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('25-APR-2020', 'DD-MON-YYYY'), 'Scurgere de gaze', 'Finalizata');
```

```sql
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('01-OCT-2021', 'DD-MON-YYYY'), 'Teava de apa sparta', 'In lucuru');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('12-DEC-2022', 'DD-MON-YYYY'), 'Scurgere de gaze', 'Neinceputa');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('25-NOV-2025', 'DD-MON-YYYY'), 'Teava de gaze sparte', 'Finalizata');
INSERT INTO Interventii (cod_interventie, data_incepere, nume_defectiune)
VALUES (Secventa.nextval, TO_DATE('15-MAY-2022', 'DD-MON-YYYY'), 'Cablu de electricitate intrerupt');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('20-SEP-2021', 'DD-MON-YYYY'), 'Internet picat', 'Finalizata');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('05-MAY-2022', 'DD-MON-YYYY'), 'Apa nefunctionala', 'In lucru');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('19-APR-2021', 'DD-MON-YYYY'), 'Siguranta sarita', 'In lucru');
INSERT INTO Interventii (cod_interventie, data_incepere, nume_defectiune)
VALUES (Secventa.nextval, TO_DATE('20-JUN-2025', 'DD-MON-YYYY'), 'Teava de apa fisurata');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('23-APR-2023', 'DD-MON-YYYY'), 'Siguranta sarita', 'Neinceputa');
INSERT INTO Interventii
VALUES (Secventa.nextval, TO_DATE('25-APR-2020', 'DD-MON-YYYY'), 'Scurgere de gaze', 'Finalizata');

SELECT *
FROM Interventii;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.001 seconds

	COD_INTERVENTIE	DATA_INCEPERE	NUME_DEFECTIUNE	STATUS
1	1	01-OCT-21	Teava de apa sparta	In lucuru
2	2	12-DEC-22	Scurgere de gaze	Neinceputa
3	3	25-NOV-25	Teava de gaze sparte	Finalizata
4	4	15-MAY-22	Cablu de electricitate intrerupt	Neinceputa
5	5	20-SEP-21	Internet picat	Finalizata
6	6	05-MAY-22	Apa nefunctionala	In lucru
7	7	19-APR-21	Siguranta sarita	In lucru
8	8	20-JUN-25	Teava de apa fisurata	Neinceputa
9	9	23-APR-23	Siguranta sarita	Neinceputa
10	10	25-APR-20	Scurgere de gaze	Finalizata

- DOCUMENTE

```sql
INSERT INTO Documente
VALUES (1, 3, 6, 4, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 3, 2, 4, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 3, 5, 4, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 9, 6, 9, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (1, 10, 1, 1, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (1, 10, 6, 6, 'Dumitrescu', 'Marian');
```

```
INSERT INTO Documente
VALUES (1, 10, 3, 6, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 10, 10, 6, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 1, 9, 3, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 9, 1, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 9, 6, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 9, 3, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 6, 1, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 6, 6, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 6, 3, 8, 'Andrei', 'Constantin');
```

Worksheet | Query Builder

```
VALUES (1, 3, 2, 4, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 3, 5, 4, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 9, 6, 9, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (1, 10, 1, 1, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (1, 10, 6, 6, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 10, 3, 6, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 10, 10, 6, 'Dumitrescu', 'Marian');
INSERT INTO Documente
VALUES (1, 1, 9, 3, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 9, 1, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 9, 6, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 9, 3, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 6, 1, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 6, 6, 8, 'Andrei', 'Constantin');
INSERT INTO Documente
VALUES (2, 6, 3, 8, 'Andrei', 'Constantin');

SELECT *
FROM Documente;
```

Script Output | Query Result

SQL | All Rows Fetched: 15 in 0.003 seconds

	COD_PLANGERE	COD_CLIENT	COD_MUNCITOR	COD_INTERVENTIE	NUME_SEMNATAR	PRENUME_SEMNATAR
4	1	9	6	9	Andrei	Constantin
5	1	10	1	1	Andrei	Constantin
6	1	10	6	6	Dumitrescu	Marian
7	1	10	3	6	Dumitrescu	Marian
8	1	10	10	6	Dumitrescu	Marian
9	1	1	9	3	Andrei	Constantin
10	2	9	1	8	Andrei	Constantin
11	2	9	6	8	Andrei	Constantin
12	2	9	3	8	Andrei	Constantin
13	2	6	1	8	Andrei	Constantin
14	2	6	6	8	Andrei	Constantin
15	2	6	3	8	Andrei	Constantin

6. Dându-se un array ce contine orașele "Chișinău", "Craiova", "București", "Iași", "Bacău", "Kiev", "Beijing" să se afișeze pentru fiecare client dacă are cel puțin o adresa în cel puțin un oraș care se regăsește în tabelul Adrese, dar nu se regăsește printre cele date. Pentru fiecare client, pentru fiecare astfel de oraș sa se afiseze emailul și orașul respectiv, iar dacă nu exista niciun astfel de oraș sa se afișeze "Nu exista niciun oraș".

```
CREATE OR REPLACE PROCEDURE cerinta_6 IS
  TYPE vector_orase IS VARRAY(20) OF Adrese.oras%TYPE;
  TYPE tabel_orase IS TABLE OF Adrese.oras%TYPE INDEX BY PLS_INTEGER;
  TYPE tabel_clienti IS TABLE OF clienti.email%TYPE INDEX BY PLS_INTEGER;
  v_vector_orase vector_orase := vector_orase('Chisinau', 'Craiova', 'Bucuresti', 'Iasi', 'Bacau', 'Kiev',
'Beijing');
  v_tabel_orase tabel_orase;
  v_tabel_clienti tabel_clienti;     --retine id-ul pt un client
  v_numar_adrese NUMBER(4);
  v_minim_o_adresa BOOLEAN;
  v_email Clienti.email%TYPE;
  j NUMBER(4);
BEGIN

    SELECT DISTINCT oras          --punem toate orasele in v_tabel_orase
    BULK COLLECT INTO v_tabel_orase
    FROM Adrese;

    FOR i IN v_vector_orase.FIRST..v_vector_orase.LAST LOOP      --scoatem din v_tabel_orase
orasele care sunt in v_vector_orase
      j := v_tabel_orase.FIRST;
      WHILE(j IS NOT NULL) LOOP
        IF v_vector_orase(i) = v_tabel_orase(j) THEN
          v_tabel_orase.DELETE(j);
          EXIT;
        END IF;
        j := v_tabel_orase.NEXT(j);
      END LOOP;
    END LOOP;

    SELECT cod_client
    BULK COLLECT INTO v_tabel_clienti
    FROM Clienti;

    FOR i IN v_tabel_clienti.FIRST..v_tabel_clienti.LAST LOOP
      v_minim_o_adresa := FALSE;
      j := v_tabel_orase.FIRST;
```

```
        SELECT email
        INTO v_email
        FROM Clienti
        WHERE cod_client = v_tabel_clienti(i);

        WHILE(j IS NOT NULL) LOOP
          v_numar_adrese := 0;

           SELECT NVL2((SELECT COUNT(*)
                    FROM Adrese
                    WHERE cod_client = v_tabel_clienti(i)
                    AND oras = v_tabel_orase(j)
                    GROUP BY oras), (SELECT COUNT(*)
                              FROM Adrese
                              WHERE cod_client = v_tabel_clienti(i)
                              AND oras = v_tabel_orase(j)
                              GROUP BY oras), 0)
          INTO v_numar_adrese
          FROM dual;

          IF v_numar_adrese > 0 THEN
            DBMS_OUTPUT.PUT_LINE('Clientul cu mailul ' || v_email || ' are adresa in orasul ' ||
v_tabel_orase(j));
            v_minim_o_adresa := TRUE;
          END IF;

          j := v_tabel_orase.NEXT(j);
        END LOOP;

        IF v_minim_o_adresa = FALSE THEN
          DBMS_OUTPUT.PUT_LINE('Clientul cu mailul ' || v_email || ' nu are nicio adresa in orasele
date');
        END IF;
      END LOOP;
END;


DECLARE
BEGIN
   cerinta_6;
END;
```

Worksheet | Query Builder

```
--6
CREATE OR REPLACE PROCEDURE cerinta_6 IS
   TYPE vector_orase IS VARRAY(20) OF Adrese.oras%TYPE;
   TYPE tabel_orase IS TABLE OF Adrese.oras%TYPE INDEX BY PLS_INTEGER;
   TYPE tabel_clienti IS TABLE OF clienti.email%TYPE INDEX BY PLS_INTEGER;
   v_vector_orase vector_orase := vector_orase('Chisinau', 'Craiova', 'Bucuresti', 'Iasi', 'Bacau', 'Kiev', 'Beijing');
   v_tabel_orase tabel_orase;
   v_tabel_clienti tabel_clienti;     --retine id-ul pt un client
   v_numar_adrese NUMBER(4);
   v_minim_o_adresa BOOLEAN;
   v_email Clienti.email%TYPE;
   j NUMBER(4);
BEGIN

      SELECT DISTINCT oras          --punem toate orasele in v_tabel_orase
      BULK COLLECT INTO v_tabel_orase
      FROM Adrese;

      FOR i IN v_vector_orase.FIRST..v_vector_orase.LAST LOOP      --scoatem din v_tabel_orase orasele care sunt in v_vector_orase
         j := v_tabel_orase.FIRST;
         WHILE(j IS NOT NULL) LOOP
            IF v_vector_orase(i) = v_tabel_orase(j) THEN
               v_tabel_orase.DELETE(j);
               EXIT;
            END IF;
            j := v_tabel_orase.NEXT(j);
         END LOOP;
      END LOOP;

      SELECT cod_client
      BULK COLLECT INTO v_tabel_clienti
      FROM Clienti;

      FOR i IN v_tabel_clienti.FIRST..v_tabel_clienti.LAST LOOP
         v_minim_o_adresa := FALSE;
         j := v_tabel_orase.FIRST;
```

Query Result | Script Output

Task completed in 0.319 seconds

```
Procedure CERINTA_6 compiled


PL/SQL procedure successfully completed.
```

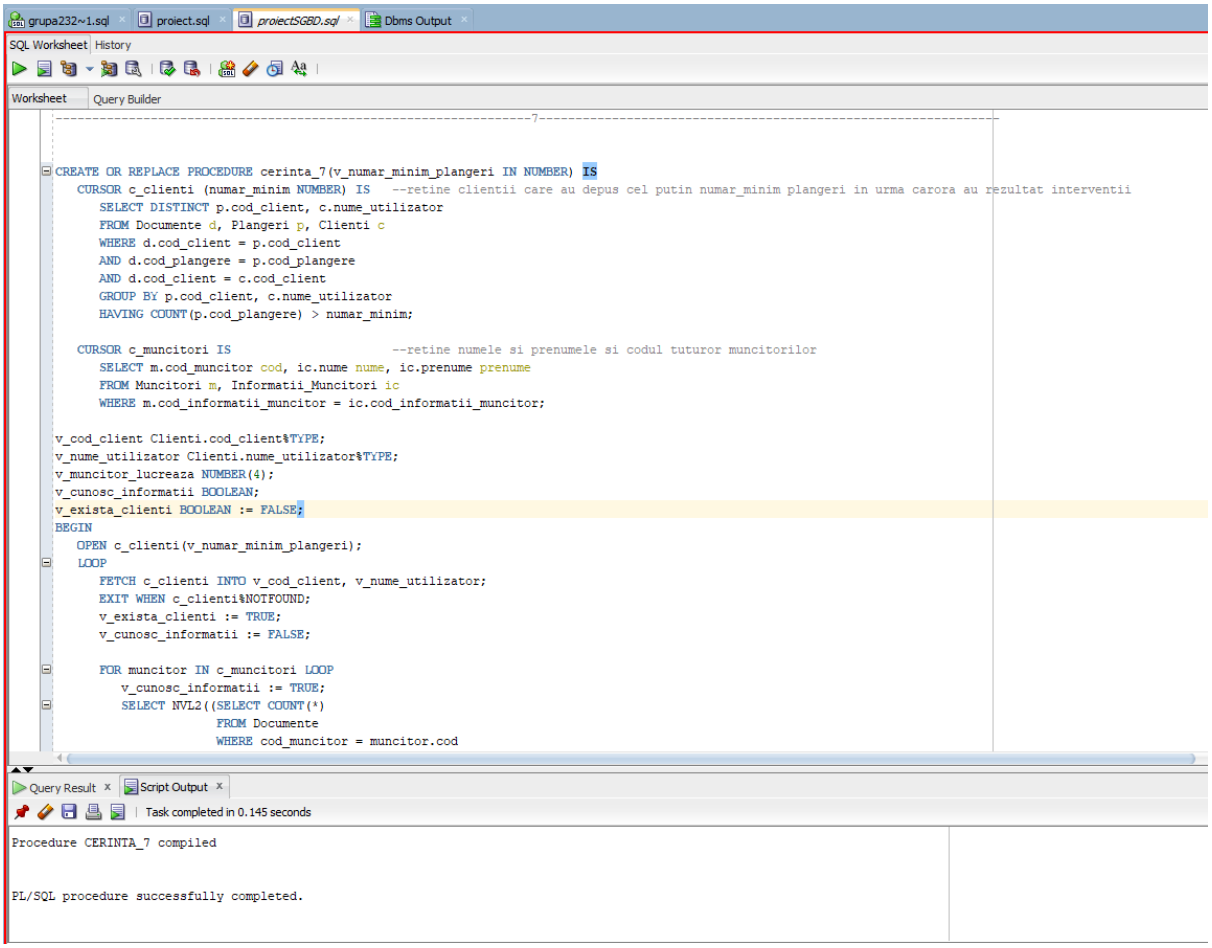
```
Clientul cu mailul marian08@yahoo.com are adresa in orasul Amsterdam
Clientul cu mailul vasy_john@outlook.ro are adresa in orasul Arad
Clientul cu mailul criss@gmail.com nu are nicio adresa in orasele date
Clientul cu mailul buzion23@s.unibuc.ro are adresa in orasul Timisioara
Clientul cu mailul ana@gmail.com nu are nicio adresa in orasele date
Clientul cu mailul ion004@yahoo.com nu are nicio adresa in orasele date
Clientul cu mailul dragos244@s.unibuc.ro nu are nicio adresa in orasele date
Clientul cu mailul seb@yahoo.com are adresa in orasul Paris
Clientul cu mailul maria_andreea@gmail.ro are adresa in orasul Buzau
Clientul cu mailul frone34@outlook.com are adresa in orasul Londra
Clientul cu mailul frone34@outlook.com are adresa in orasul Constanta
Clientul cu mailul frone34@outlook.com are adresa in orasul Cluj
Clientul cu mailul mexicanu@outlook.com are adresa in orasul Botosani
Clientul cu mailul matei@outlook.com are adresa in orasul Vaslui
Clientul cu mailul test@yahoo nu are nicio adresa in orasele date
Clientul cu mailul test2@yahoo nu are nicio adresa in orasele date
Clientul cu mailul test3@yahoo nu are nicio adresa in orasele date
```

7. Să se afișeze numele și prenumele muncitorilor care sunt desemnati sa intervina la interventii survenite în urma plangerilor depuse de clienți care au cel puțin o plângere depusă în urma careia au fost intervenții (sau urmează sa fie), cât și numele de utilizator al acestor clienți. Dacă despre un anumit muncitor nu se cunosc informații (numele și prenumele) nu se va afișa nimic. Dacă pentru un client nu se cunosc informații despre niciunul dintre muncitorii desemnați să-i rezolve plangerile, sa se precizeze acest lucru. Dacă nu exista niciun client care sa fi depus cel puțin o plangere în urma cărora sa existe intervenții, sa se precizeze acest lucru.

```
CREATE OR REPLACE PROCEDURE cerinta_7(v_numar_minim_plangeri IN NUMBER) IS
  CURSOR c_clienti (numar_minim NUMBER) IS   --retine clientii care au depus cel putin
numar_minim plangeri in urma carora au rezultat interventii
    SELECT DISTINCT p.cod_client, c.nume_utilizator
    FROM Documente d, Plangeri p, Clienti c
    WHERE d.cod_client = p.cod_client
    AND d.cod_plangere = p.cod_plangere
    AND d.cod_client = c.cod_client
    GROUP BY p.cod_client, c.nume_utilizator
    HAVING COUNT(p.cod_plangere) > numar_minim;

  CURSOR c_muncitori IS                    --retine numele si prenumele si codul tuturor muncitorilor
    SELECT m.cod_muncitor cod, ic.nume nume, ic.prenume prenume
```

```
    FROM Muncitori m, Informatii_Muncitori ic
    WHERE m.cod_informatii_muncitor = ic.cod_informatii_muncitor;

v_cod_client Clienti.cod_client%TYPE;
v_nume_utilizator Clienti.nume_utilizator%TYPE;
v_muncitor_lucreaza NUMBER(4);
v_cunosc_informatii BOOLEAN;
v_exista_clienti BOOLEAN := FALSE;
BEGIN
  OPEN c_clienti(v_numar_minim_plangeri);
  LOOP
    FETCH c_clienti INTO v_cod_client, v_nume_utilizator;
    EXIT WHEN c_clienti%NOTFOUND;
    v_exista_clienti := TRUE;
    v_cunosc_informatii := FALSE;

    FOR muncitor IN c_muncitori LOOP
      v_cunosc_informatii := TRUE;
      SELECT NVL2((SELECT COUNT(*)
                FROM Documente
                WHERE cod_muncitor = muncitor.cod
                AND cod_client = v_cod_client), (SELECT COUNT(*)
                                FROM Documente
                                WHERE cod_muncitor = muncitor.cod
                                AND cod_client = v_cod_client), 0)
      INTO v_muncitor_lucreaza
      FROM dual;

      IF v_muncitor_lucreaza > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Pentru clientul ' || v_nume_utilizator || ' lucreaza muncitorul ' ||
muncitor.nume || ' ' ||  muncitor.prenume);
      END IF;
    END LOOP;

    IF v_cunosc_informatii = FALSE THEN
      DBMS_OUTPUT.PUT_LINE('Nu se cunoaste numele si prenumele niciunuia dintre muncitorii
care au desemnati sa rezolve plangerile clientului ' || v_nume_utilizator);
    END IF;
  END LOOP;

  CLOSE c_clienti;

  IF v_exista_clienti = FALSE THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nu exista niciun client care sa depuna minim ' ||
v_numar_minim_plangeri || ' plangeri in urma carora sa exista interventii');
  END IF;
END;


DECLARE
BEGIN
  cerinta_7(1);
END;
```



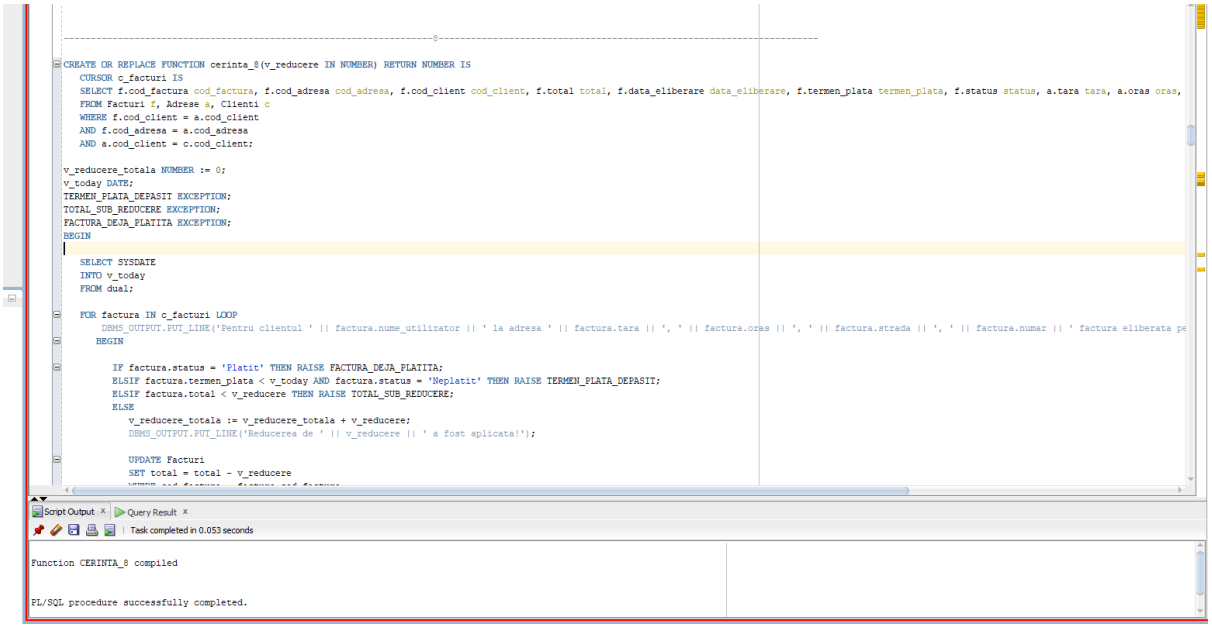
```
grupa232~1.sql   proiect.sql   proiectSGBD.sql   Dbms Output
Buffer Size: 20000
proiectSGBD
Pentru clientul Icris lucreaza muncitorul Mincu Ionut
Pentru clientul Icris lucreaza muncitorul Loredana Mariana
Pentru clientul Icris lucreaza muncitorul Ion Antonescu
Pentru clientul dimaI lucreaza muncitorul Ion Andrei
Pentru clientul dimaI lucreaza muncitorul Loredana Mariana
Pentru clientul andreeaMaria lucreaza muncitorul Ion Andrei
Pentru clientul andreeaMaria lucreaza muncitorul Loredana Mariana
Pentru clientul kingFrone lucreaza muncitorul Ion Andrei
Pentru clientul kingFrone lucreaza muncitorul Ioana Maria
Pentru clientul kingFrone lucreaza muncitorul Loredana Mariana
```

8. Să se reducă totalul facturilor clienților cu 'v reducere'(parametru al funcției) și sa se afiseza reducerea totala a facturilor. Dacă factură a fost deja plătită nu se va intampla nimic. Dacă termenul de plata a fost depășit, iar factura nu a fost plătită, totalul nu se va reduce, dar termenul de plata va fi extins pentru 1 Ianuarie 2024. Dacă termenul de plata nu a fost depasit, dar totalul facturii este mai mic decat reducerea, atunci statusul facturii va deveni 'Platit'.

```
CREATE OR REPLACE FUNCTION cerinta_8(v_reducere IN NUMBER) RETURN NUMBER IS
  CURSOR c_facturi IS
  SELECT f.cod_factura cod_factura, f.cod_adresa cod_adresa, f.cod_client cod_client, f.total total,
f.data_eliberare data_eliberare, f.termen_plata termen_plata, f.status status, a.tara tara, a.oras oras,
a.strada strada, a.numar numar, c.nume_utilizator nume_utilizator
  FROM Facturi f, Adrese a, Clienti c
  WHERE f.cod_client = a.cod_client
  AND f.cod_adresa = a.cod_adresa
  AND a.cod_client = c.cod_client;

v_reducere_totala NUMBER := 0;
v_today DATE;
TERMEN_PLATA_DEPASIT EXCEPTION;
TOTAL_SUB_REDUCERE EXCEPTION;
FACTURA_DEJA_PLATITA EXCEPTION;
BEGIN

  SELECT SYSDATE
```

```
  INTO v_today
  FROM dual;

  FOR factura IN c_facturi LOOP
     DBMS_OUTPUT.PUT_LINE('Pentru clientul ' || factura.nume_utilizator || ' la adresa ' ||
factura.tara || ', ' || factura.oras || ', ' || factura.strada || ', ' || factura.numar || ' factura eliberata pe data de '
|| factura.data_eliberare || ': ');
    BEGIN

      IF factura.status = 'Platit' THEN RAISE FACTURA_DEJA_PLATITA;
      ELSIF factura.termen_plata < v_today AND factura.status = 'Neplatit' THEN RAISE
TERMEN_PLATA_DEPASIT;
      ELSIF factura.total < v_reducere THEN RAISE TOTAL_SUB_REDUCERE;
      ELSE
        v_reducere_totala := v_reducere_totala + v_reducere;
        DBMS_OUTPUT.PUT_LINE('Reducerea de ' || v_reducere || ' a fost aplicata!');

        UPDATE Facturi
        SET total = total - v_reducere
        WHERE cod_factura = factura.cod_factura
        AND cod_adresa = factura.cod_adresa
        AND cod_client = factura.cod_client;
      END IF;

      EXCEPTION

        WHEN TERMEN_PLATA_DEPASIT THEN
          DBMS_OUTPUT.PUT_LINE('Termenul de plata a fost depasit! Totalul nu a fost redus, inse
termenul de plata a fost extins pentru 1 Ianuarie 2024');

          UPDATE Facturi
          SET termen_plata = TO_DATE('01-JAN-2024', 'DD-MON-YYYY')
          WHERE cod_factura = factura.cod_factura
          AND cod_adresa = factura.cod_adresa
          AND cod_client = factura.cod_client;

        WHEN TOTAL_SUB_REDUCERE THEN
          v_reducere_totala := v_reducere_totala + factura.total;
          DBMS_OUTPUT.PUT_LINE('Totalul de plata este mai mic decat reducerea, asa ca factura a
fost actualizata ca fiind platita');

          UPDATE Facturi
          SET status = 'Platit'
          WHERE cod_factura = factura.cod_factura
```

```
            AND cod_adresa = factura.cod_adresa
            AND cod_client = factura.cod_client;

          WHEN FACTURA_DEJA_PLATITA THEN
            DBMS_OUTPUT.PUT_LINE('Factura a fost deja platita');
      END;
      DBMS_OUTPUT.PUT_LINE(' ');
    END LOOP;

    RETURN v_reducere_totala;
END;


DECLARE                          --aici sunt evidentiate toate cazurile
v_reducere_totala NUMBER;
BEGIN
   v_reducere_totala := cerinta_8(500);
   DBMS_OUTPUT.PUT_LINE(' ');
   DBMS_OUTPUT.PUT_LINE('Reducere totala: ' || v_reducere_totala);
END;
```



```
grupa232~1.sql   proiect.sql   proiectSGBD.sql   Dbms Output
Buffer Size: 20000
proiectSGBD
Factura a fost deja platita

Pentru clientul dragos244 la adresa Republica Moldova, Chisinau, Strada Mosilor, 234 factura eliberata pe data de 20-AUG-23:
Factura a fost deja platita

Pentru clientul kingFrone la adresa Romania, Constanta, Faleza Marii, 634 factura eliberata pe data de 14-JAN-23:
Factura a fost deja platita

Pentru clientul kingFrone la adresa Romania, Constanta, Faleza Marii, 634 factura eliberata pe data de 27-DEC-24:
Factura a fost deja platita

Pentru clientul kingFrone la adresa Romania, Constanta, Faleza Marii, 634 factura eliberata pe data de 03-APR-23:
Factura a fost deja platita

Pentru clientul john_vasy la adresa Romania, Arad, Strda Aradului, 65 factura eliberata pe data de 24-SEP-20:
Factura a fost deja platita

Pentru clientul andreeaMaria la adresa Romania, Buzau, Strda Buzaului, 23 factura eliberata pe data de 14-DEC-22:
Factura a fost deja platita

Pentru clientul mexicanu la adresa Romania, Botosani, Strda Botosani, 653 factura eliberata pe data de 05-FEB-22:
Factura a fost deja platita

Pentru clientul MAtei321 la adresa Romania, Vaslui, Strda Vslui, 13 factura eliberata pe data de 19-APR-21:
Factura a fost deja platita

Pentru clientul TestUsername la adresa Romania, Craiova, Stadionului, 80 factura eliberata pe data de 12-DEC-20:
Reducerea de 500 a fost aplicata!

Pentru clientul TestUsername2 la adresa Romania, Craiova, Stadionului, 80 factura eliberata pe data de 12-DEC-20:
Factura a fost deja platita

Pentru clientul TestUsername2 la adresa Romania, Craiova, Stadionului, 80 factura eliberata pe data de 01-APR-19:
Factura a fost deja platita

Pentru clientul TestUsername3 la adresa Romania, Craiova, Stadionului, 80 factura eliberata pe data de 12-DEC-20:
Reducerea de 500 a fost aplicata!

Pentru clientul TestUsername2 la adresa Romania, Craiova, Stadionului, 80 factura eliberata pe data de 12-DEC-20:
Reducerea de 500 a fost aplicata!

Pentru clientul TestUsername2 la adresa Romania, Craiova, Stadionului, 80 factura eliberata pe data de 01-APR-19:
Reducerea de 500 a fost aplicata!


Reducere totala: 2000
```

9. Dându-se un nume și un prenume ca parametrii ai procedurii, sa se gaseasca clientul, sa se actualizeze toate facturile sale ca fiind plătite, și sa se afiseze cardurile (numărul și codul de securitate al acestora) deținute de acest client, dacă factura pe numele lui emisă pe România (dacă sunt mai multe astfel de facturi cel puțin una trebuie sa aiba statusul 'Neplatit') este Neplatita și soldul contului mai mare de 2000. (condiția de dacă se aplica doar pt afișarea cardurilor, nu și pt actualizarea facturilor)

```
CREATE OR REPLACE PROCEDURE cerinta_9(v_nume Informatii_Clienti.nume%TYPE,
v_prenume Informatii_Clienti.prenume%TYPE) IS
CURSOR c_carduri (cod_client_param Clienti.cod_client%TYPE) IS
    SELECT DISTINCT cb.numar_card, cb.cod_securitate_card
    FROM Carduri_Bancare cb, Informatii_Bancare ib, Clienti c, Informatii_Clienti ic, Adrese a,
Facturi f
    WHERE cb.cod_client = cod_client_param
```

```
    AND ib.cod_client = cod_client_param
    AND ib.sold_curent > 2000
    AND a.cod_client = cod_client_param
    AND a.tara = 'Romania'
    AND f.cod_client = cod_client_param
    AND f.cod_adresa = a.cod_adresa
    AND f.status = 'Neplatit';

v_cod_client Clienti.cod_client%TYPE;
v_numar_card Carduri_bancare.numar_card%TYPE;
v_cod_securitate_card Carduri_bancare.cod_securitate_card%TYPE;
v_index_card NUMBER(4) := 1;
BEGIN

  SELECT c.cod_client      --aflare cod client
  INTO v_cod_client
  FROM Informatii_Clienti ic, Clienti c
  WHERE c.cod_informatii_client = ic.cod_informatii_client
  AND ic.nume = v_nume
  AND ic.prenume = v_prenume;

  OPEN c_carduri(v_cod_client);
  LOOP
    FETCH c_carduri INTO v_numar_card, v_cod_securitate_card;
    EXIT WHEN c_carduri%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Cardul ' || v_index_card || ': numar - ' || v_numar_card || ', cod
securitate - ' || v_cod_securitate_card);
    v_index_card := v_index_card + 1;
  END LOOP;

  UPDATE Facturi
  SET status = 'Platit'
  WHERE cod_client = v_cod_client;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE ('NO DATA FOUND!');
    WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE ('PREA MULTE
REZULTATE!');
    WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE ('ALTA EROARE!');
END;
```

```
DECLARE   -- no data found; din cauza ca nu exista niciun client cu numele si prenumele asta
BEGIN
  cerinta_9('AAAA', 'VVVV');
END;

DECLARE   -- to many rows; din cauza ca exista 2 clienti cu numele si prenumele astea
BEGIN
  cerinta_9('Ion', 'Vasile');
END;


DECLARE   -- no data found; din cauza ca desi exista o inregistrare in Informatii_Clienti cu numele si
prenumele date, acea inregistrare nu este legata de niciun client din tabelul Clienti
BEGIN
  cerinta_9('Gigel', 'Frone');
END;

DECLARE   -- se afieaza cum trebuie
BEGIN
  cerinta_9('NumeTest', 'PrenumeTest');
END;

DECLARE   -- se afieaza cum trebuie desi sunt 2 facturi neplatite pe adrese de romania; se selecteaza
ambele dar pentru ca folosim select distinct cardurile sunt luat o singura data
BEGIN
  cerinta_9('NumeTest2', 'PrenumeTest2');
END;

DECLARE   -- nu se afiseaza niciun card pt ca nu exista, dar se executa ok fara sa dea no data found
BEGIN
  cerinta_9('NumeTest3', 'PrenumeTest3');
END;
```

```
CREATE OR REPLACE PROCEDURE cerinta_9(v_nume Informatii_Clienti.nume%TYPE, v_prenume Informatii_Clienti.prenume%TYPE) IS
CURSOR c_carduri (cod_client_param Clienti.cod_client%TYPE) IS
      SELECT DISTINCT cb.numar_card, cb.cod_securitate_card
      FROM Carduri_Bancare cb, Informatii_Bancare ib, Clienti c, Informatii_Clienti ic, Adrese a, Facturi f
      WHERE cb.cod_client = cod_client_param
      AND ib.cod_client = cod_client_param
      AND ib.sold_curent > 2000
      AND a.cod_client = cod_client_param
      AND a.tara = 'Romania'
      AND f.cod_client = cod_client_param
      AND f.cod_adresa = a.cod_adresa
      AND f.status = 'Neplatit';

v_cod_client Clienti.cod_client%TYPE;
v_numar_card Carduri_bancare.numar_card%TYPE;
v_cod_securitate_card Carduri_bancare.cod_securitate_card%TYPE;
v_index_card NUMBER(4) := 1;
BEGIN

   SELECT c.cod_client      --aflare cod client
   INTO v_cod_client
   FROM Informatii_Clienti ic, Clienti c
   WHERE c.cod_informatii_client = ic.cod_informatii_client
   AND ic.nume = v_nume
   AND ic.prenume = v_prenume;

   OPEN c_carduri(v_cod_client);
   LOOP
      FETCH c_carduri INTO v_numar_card, v_cod_securitate_card;
      EXIT WHEN c_carduri%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Cardul ' || v_index_card || ': numar - ' || v_numar_card || ', cod securitate - ' || v_cod_securitate_card);
```

Script Output | Query Result

Task completed in 0.044 seconds

```
Procedure CERINTA_9 compiled


PL/SQL procedure successfully completed.
```

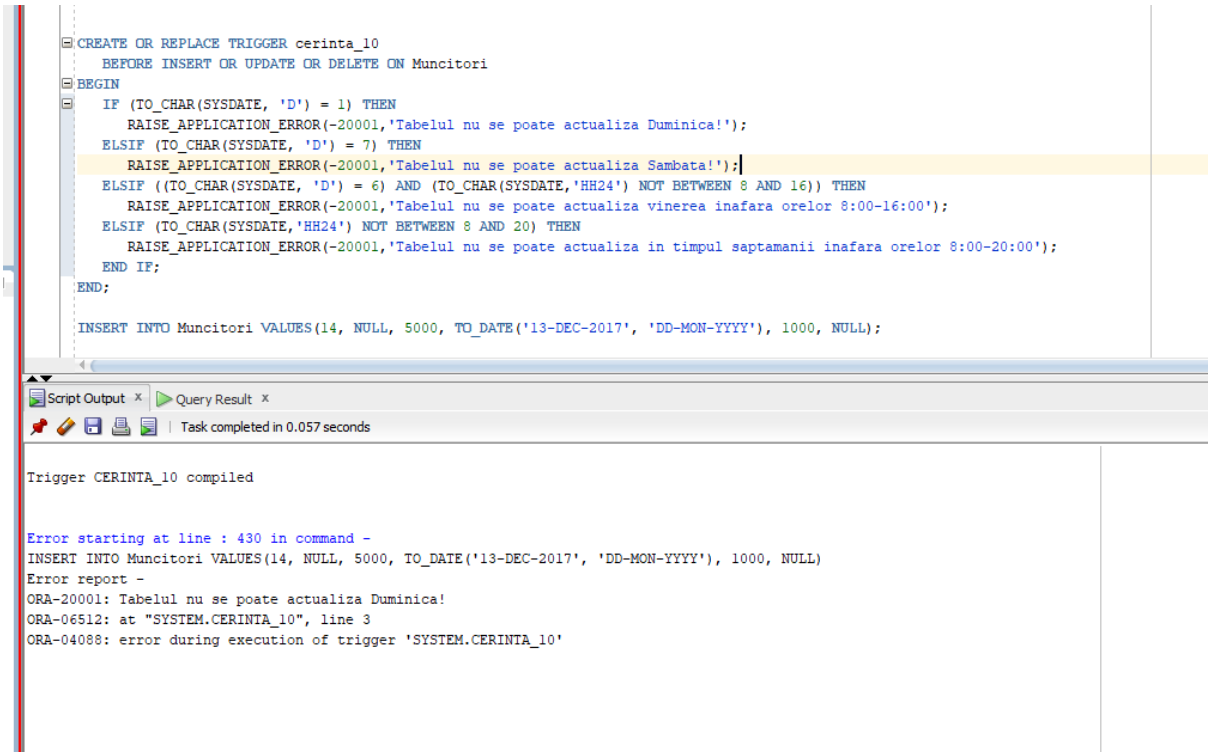
Compiler - Log

```
NO DATA FOUND!

PREA MULTE REZULTATE!

Cardul 1: numar - 1111111111111111, cod securitate - 111
Cardul 2: numar - 2222222222222222, cod securitate - 222
Cardul 3: numar - 3333333333333333, cod securitate - 222
```

10. Definiți un trigger care sa nu permita angajarea, concedierea sau modificarea contractelor de muncă a muncitorilor (adică sa nu permita actualizari asupra tabelei Muncitori) în afara programului de lucru, acesta fiind Luni-Joi între 8:00 - 20:00 și Vineri între 8:00 - 16:00.

```
CREATE OR REPLACE TRIGGER cerinta_10
  BEFORE INSERT OR UPDATE OR DELETE ON Muncitori
BEGIN
  IF (TO_CHAR(SYSDATE, 'D') = 7) THEN
    RAISE_APPLICATION_ERROR(-20001,'Tabelul nu se poate actualiza Duminica!');
  ELSIF (TO_CHAR(SYSDATE, 'D') = 6) THEN
    RAISE_APPLICATION_ERROR(-20001,'Tabelul nu se poate actualiza Sambata!');
```

```
  ELSIF ((TO_CHAR(SYSDATE, 'D') = 5) AND (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN
8 AND 16)) THEN
    RAISE_APPLICATION_ERROR(-20001,'Tabelul nu se poate actualiza vinerea inafara orelor
8:00-16:00');
  ELSIF (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 20) THEN
    RAISE_APPLICATION_ERROR(-20001,'Tabelul nu se poate actualiza in timpul saptamanii
inafara orelor 8:00-20:00');
  END IF;
END;

INSERT INTO Muncitori VALUES(14, NULL, 5000, TO_DATE('13-DEC-2017',
'DD-MON-YYYY'), 1000);
```



11. Definiți un trigger care sa nu permita plata facturilor (actualizarea lor ca fiind 'Platit') dacă totalul acestora este mai mare decat soldul curent al clientului. Dacă clientul nu are setate informațiile bancare (nu exista în Informatii_Clienti nicio inserare care sa aiba cod_client al celui căruia îi aparține factura), se va crea o excepție.

```
CREATE OR REPLACE TRIGGER cerinta_11
  BEFORE UPDATE OF status ON Facturi
  FOR EACH ROW
DECLARE
v_sold_curent Informatii_Bancare.sold_curent%TYPE;
```

```
BEGIN

  IF :NEW.status = 'Platit' THEN
    SELECT sold_curent
    INTO v_sold_curent
    FROM Informatii_Bancare
    WHERE cod_client = :NEW.cod_client;

    IF v_sold_curent < :NEW.total THEN
      RAISE_APPLICATION_ERROR(-20001, 'Clientul nu dispune de suficienti bani pentru a plati
factura!');
    END IF;
  END IF;

  EXCEPTION   --pentru cazul in care clientul nu are inserare in informatii_bancare
    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001, 'Clientul nu are
informatiile bancare setate!');
END;

UPDATE Facturi SET status = 'Platit' WHERE cod_client = 15; --ne trimite in exceptie pentru ca nu
exista in tableul Informatii_Bancare un client cu codul 15
UPDATE Facturi SET status = 'Platit' WHERE cod_client = 10;    --exista inserare in
Informatii_Clienti pentru clientul 10, dar nu dispune de suficienti bani
```

```
CREATE OR REPLACE TRIGGER cerinta_11
   BEFORE UPDATE OF status ON Facturi
   FOR EACH ROW
DECLARE
v_sold_curent Informatii_Bancare.sold_curent%TYPE;
BEGIN

   IF :NEW.status = 'Platit' THEN
      SELECT sold_curent
      INTO v_sold_curent
      FROM Informatii_Bancare
      WHERE cod_client = :NEW.cod_client;

      IF v_sold_curent < :NEW.total THEN
         RAISE_APPLICATION_ERROR(-20001, 'Clientul nu dispune de suficienti bani pentru a plati factura!');
      END IF;
    END IF;

    EXCEPTION   --pentru cazul in care clientul nu are inserare in informatii_bancare
      WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20001, 'Clientul nu are informatiile bancare setate!');
END;
```

Script Output | Query Result

Task completed in 0.041 seconds

```
Trigger CERINTA_11 compiled


Error starting at line : 462 in command -
UPDATE Facturi SET status = 'Platit' WHERE cod_client = 10
Error report -
ORA-20001: Clientul nu dispune de suficienti bani pentru a plati factura!
ORA-06512: at "SYSTEM.CERINTA_11", line 12
ORA-04088: error during execution of trigger 'SYSTEM.CERINTA_11'


Error starting at line : 461 in command -
UPDATE Facturi SET status = 'Platit' WHERE cod_client = 15
Error report -
ORA-20001: Clientul nu are informatiile bancare setate!
ORA-06512: at "SYSTEM.CERINTA_11", line 17
ORA-04088: error during execution of trigger 'SYSTEM.CERINTA_11'
```

Compiler - Log

Messages | Logging Page | Statements | Compiler

12. Definiți un trigger care sa permita crearea, ștergerea sau modificarea structurii tabelelor doar utilizatorului 'SYSTEM'.

```
CREATE OR REPLACE TRIGGER cerinta_12
  BEFORE CREATE OR DROP OR ALTER ON SCHEMA
DECLARE
v_operation VARCHAR(30) := SYS.SYSEVENT;
v_table_name VARCHAR2(30) := SYS.DICTIONARY_OBJ_NAME;
v_user VARCHAR(50) := SYS.LOGIN_USER;
BEGIN
  IF v_user != 'SYSTEM' THEN
    IF v_operation = 'CREATE' THEN
      RAISE_APPLICATION_ERROR(-20001, 'Nu aveti dreptul sa creati tabele noi! Tabelul ' ||
v_table_name || ' nu a fost creat.');
    ELSIF v_operation = 'ALTER' THEN
      RAISE_APPLICATION_ERROR(-20001, 'Nu aveti dreptul sa modificati tabele! Tabelul ' ||
v_table_name || ' nu a fost modificat.');
    ELSIF v_operation = 'DROP' THEN
      RAISE_APPLICATION_ERROR(-20001, 'Nu aveti dreptul sa stergeti tabele! Tabelul ' ||
v_table_name || ' nu a fost sters.');
```

```
    END IF;
  END IF;
END;
```

CREATE TABLE test (utilizator VARCHAR2(30), nume_bd VARCHAR2(50), eveniment VARCHAR2(20), nume_obiect VARCHAR2(30), data DATE);
ALTER TABLE Muncitori ADD TestColumn VARCHAR2(50);
DROP TABLE Clienti;

```
--------------------------------------------------------------------------------12--------------------------------------------------------------------------------

CREATE OR REPLACE TRIGGER cerinta_12
   BEFORE CREATE OR DROP OR ALTER ON SCHEMA
DECLARE
v_operation VARCHAR(30) := SYS.SYSEVENT;
v_table_name VARCHAR2(30) := SYS.DICTIONARY_OBJ_NAME;
v_user VARCHAR(50) := SYS.LOGIN_USER;
BEGIN
   IF v_user != 'SYSTEM2' THEN
      IF v_operation = 'CREATE' THEN
         RAISE_APPLICATION_ERROR(-20001, 'Nu aveti dreptul sa creati tabele noi! Tabelul ' || v_table_name || ' nu a fost creat.');
      ELSIF v_operation = 'ALTER' THEN
         RAISE_APPLICATION_ERROR(-20001, 'Nu aveti dreptul sa modificati tabele! Tabelul ' || v_table_name || ' nu a fost modificat.');
      ELSIF v_operation = 'DROP' THEN
         RAISE_APPLICATION_ERROR(-20001, 'Nu aveti dreptul sa stergeti tabele! Tabelul ' || v_table_name || ' nu a fost sters.');
      END IF;
   END IF;
END;


CREATE TABLE test (utilizator VARCHAR2(30), nume_bd VARCHAR2(50), eveniment VARCHAR2(20), nume_obiect VARCHAR2(30), data DATE);
ALTER TABLE Muncitori ADD TestColumn VARCHAR2(50);
DROP TABLE Clienti;
```

Script Output × | Query Result ×

Task completed in 0.031 seconds

```
Trigger CERINTA_12 compiled


Error starting at line : 492 in command -
ALTER TABLE Muncitori ADD TestColumn VARCHAR2(50)
Error report -
ORA-04088: error during execution of trigger 'SYSTEM.CERINTA_12'
ORA-00604: error occurred at recursive SQL level 1
ORA-20001: Nu aveti dreptul sa modificati tabele! Tabelul MUNCITORI nu a fost modificat.
ORA-06512: at line 10
```

13.

CREATE OR REPLACE PACKAGE pachet_cerinta_13 AS
  PROCEDURE cerinta_6;
  PROCEDURE cerinta_7(v_numar_minim_plangeri IN NUMBER);
  FUNCTION cerinta_8(v_reducere IN NUMBER) RETURN NUMBER;
  PROCEDURE cerinta_9(v_nume Informatii_Clienti.nume%TYPE, v_prenume Informatii_Clienti.prenume%TYPE);
END pachet_cerinta_13;

```
CREATE OR REPLACE PACKAGE BODY pachet_cerinta_13 AS

PROCEDURE cerinta_6 IS
  TYPE vector_orase IS VARRAY(20) OF Adrese.oras%TYPE;
  TYPE tabel_orase IS TABLE OF Adrese.oras%TYPE INDEX BY PLS_INTEGER;
  TYPE tabel_clienti IS TABLE OF clienti.email%TYPE INDEX BY PLS_INTEGER;
  v_vector_orase vector_orase := vector_orase('Chisinau', 'Craiova', 'Bucuresti', 'Iasi', 'Bacau', 'Kiev',
'Beijing');
  v_tabel_orase tabel_orase;
  v_tabel_clienti tabel_clienti;     --retine id-ul pt un client
  v_numar_adrese NUMBER(4);
  v_minim_o_adresa BOOLEAN;
  v_email Clienti.email%TYPE;
  j NUMBER(4);
BEGIN

    SELECT DISTINCT oras          --punem toate orasele in v_tabel_orase
    BULK COLLECT INTO v_tabel_orase
    FROM Adrese;

    FOR i IN v_vector_orase.FIRST..v_vector_orase.LAST LOOP      --scoatem din v_tabel_orase
orasele care sunt in v_vector_orase
      j := v_tabel_orase.FIRST;
      WHILE(j IS NOT NULL) LOOP
        IF v_vector_orase(i) = v_tabel_orase(j) THEN
          v_tabel_orase.DELETE(j);
          EXIT;
        END IF;
        j := v_tabel_orase.NEXT(j);
      END LOOP;
    END LOOP;

    SELECT cod_client
    BULK COLLECT INTO v_tabel_clienti
    FROM Clienti;

    FOR i IN v_tabel_clienti.FIRST..v_tabel_clienti.LAST LOOP
      v_minim_o_adresa := FALSE;
      j := v_tabel_orase.FIRST;

      SELECT email
      INTO v_email
      FROM Clienti
```

```
        WHERE cod_client = v_tabel_clienti(i);

        WHILE(j IS NOT NULL) LOOP
          v_numar_adrese := 0;

           SELECT NVL2((SELECT COUNT(*)
                    FROM Adrese
                    WHERE cod_client = v_tabel_clienti(i)
                    AND oras = v_tabel_orase(j)
                    GROUP BY oras), (SELECT COUNT(*)
                              FROM Adrese
                              WHERE cod_client = v_tabel_clienti(i)
                              AND oras = v_tabel_orase(j)
                              GROUP BY oras), 0)
          INTO v_numar_adrese
          FROM dual;

          IF v_numar_adrese > 0 THEN
            DBMS_OUTPUT.PUT_LINE('Clientul cu mailul ' || v_email || ' are adresa in orasul ' ||
v_tabel_orase(j));
            v_minim_o_adresa := TRUE;
          END IF;

          j := v_tabel_orase.NEXT(j);
        END LOOP;

        IF v_minim_o_adresa = FALSE THEN
          DBMS_OUTPUT.PUT_LINE('Clientul cu mailul ' || v_email || ' nu are nicio adresa in orasele
date');
        END IF;
      END LOOP;
END;



PROCEDURE cerinta_7(v_numar_minim_plangeri IN NUMBER) IS
   CURSOR c_clienti (numar_minim NUMBER) IS   --retine clientii care au depus cel putin
numar_minim plangeri in urma carora au rezultat interventii
      SELECT DISTINCT p.cod_client, c.nume_utilizator
      FROM Documente d, Plangeri p, Clienti c
      WHERE d.cod_client = p.cod_client
      AND d.cod_plangere = p.cod_plangere
      AND d.cod_client = c.cod_client
      GROUP BY p.cod_client, c.nume_utilizator
```

```
    HAVING COUNT(p.cod_plangere) > numar_minim;

  CURSOR c_muncitori IS                    --retine numele si prenumele si codul tuturor muncitorilor
    SELECT m.cod_muncitor cod, ic.nume nume, ic.prenume prenume
    FROM Muncitori m, Informatii_Muncitori ic
    WHERE m.cod_informatii_muncitor = ic.cod_informatii_muncitor;

v_cod_client Clienti.cod_client%TYPE;
v_nume_utilizator Clienti.nume_utilizator%TYPE;
v_muncitor_lucreaza NUMBER(4);
v_cunosc_informatii BOOLEAN;
v_exista_clienti BOOLEAN := FALSE;
BEGIN
  OPEN c_clienti(v_numar_minim_plangeri);
  LOOP
    FETCH c_clienti INTO v_cod_client, v_nume_utilizator;
    EXIT WHEN c_clienti%NOTFOUND;
    v_exista_clienti := TRUE;
    v_cunosc_informatii := FALSE;

    FOR muncitor IN c_muncitori LOOP
      v_cunosc_informatii := TRUE;
      SELECT NVL2((SELECT COUNT(*)
                FROM Documente
                WHERE cod_muncitor = muncitor.cod
                AND cod_client = v_cod_client), (SELECT COUNT(*)
                                FROM Documente
                                WHERE cod_muncitor = muncitor.cod
                                AND cod_client = v_cod_client), 0)
      INTO v_muncitor_lucreaza
      FROM dual;

      IF v_muncitor_lucreaza > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Pentru clientul ' || v_nume_utilizator || ' lucreaza muncitorul ' ||
muncitor.nume || ' ' ||  muncitor.prenume);
      END IF;
    END LOOP;

    IF v_cunosc_informatii = FALSE THEN
      DBMS_OUTPUT.PUT_LINE('Nu se cunoaste numele si prenumele niciunuia dintre muncitorii
care au desemnati sa rezolve plangerile clientului ' || v_nume_utilizator);
    END IF;
  END LOOP;
```

```
  CLOSE c_clienti;

  IF v_exista_clienti = FALSE THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista niciun client care sa depuna minim ' ||
v_numar_minim_plangeri || ' plangeri in urma carora sa exista interventii');
  END IF;
END;



FUNCTION cerinta_8(v_reducere IN NUMBER) RETURN NUMBER IS
  CURSOR c_facturi IS
  SELECT f.cod_factura cod_factura, f.cod_adresa cod_adresa, f.cod_client cod_client, f.total total,
f.data_eliberare data_eliberare, f.termen_plata termen_plata, f.status status, a.tara tara, a.oras oras,
a.strada strada, a.numar numar, c.nume_utilizator nume_utilizator
  FROM Facturi f, Adrese a, Clienti c
  WHERE f.cod_client = a.cod_client
  AND f.cod_adresa = a.cod_adresa
  AND a.cod_client = c.cod_client;

v_reducere_totala NUMBER := 0;
v_today DATE;
TERMEN_PLATA_DEPASIT EXCEPTION;
TOTAL_SUB_REDUCERE EXCEPTION;
FACTURA_DEJA_PLATITA EXCEPTION;
BEGIN

  SELECT SYSDATE
  INTO v_today
  FROM dual;

  FOR factura IN c_facturi LOOP
    DBMS_OUTPUT.PUT_LINE('Pentru clientul ' || factura.nume_utilizator || ' la adresa ' ||
factura.tara || ', ' || factura.oras || ', ' || factura.strada || ', ' || factura.numar || ' factura eliberata pe data de '
|| factura.data_eliberare || ': ');
    BEGIN

      IF factura.status = 'Platit' THEN RAISE FACTURA_DEJA_PLATITA;
      ELSIF factura.termen_plata < v_today AND factura.status = 'Neplatit' THEN RAISE
TERMEN_PLATA_DEPASIT;
      ELSIF factura.total < v_reducere THEN RAISE TOTAL_SUB_REDUCERE;
      ELSE
        v_reducere_totala := v_reducere_totala + v_reducere;
        DBMS_OUTPUT.PUT_LINE('Reducerea de ' || v_reducere || ' a fost aplicata!');
```

```
        UPDATE Facturi
        SET total = total - v_reducere
        WHERE cod_factura = factura.cod_factura
        AND cod_adresa = factura.cod_adresa
        AND cod_client = factura.cod_client;
      END IF;

      EXCEPTION

        WHEN TERMEN_PLATA_DEPASIT THEN
          DBMS_OUTPUT.PUT_LINE('Termenul de plata a fost depasit! Totalul nu a fost redus, inse
termenul de plata a fost extins pentru 1 Ianuarie 2024');

          UPDATE Facturi
          SET termen_plata = TO_DATE('01-JAN-2024', 'DD-MON-YYYY')
          WHERE cod_factura = factura.cod_factura
          AND cod_adresa = factura.cod_adresa
          AND cod_client = factura.cod_client;

        WHEN TOTAL_SUB_REDUCERE THEN
          v_reducere_totala := v_reducere_totala + factura.total;
          DBMS_OUTPUT.PUT_LINE('Totalul de plata este mai mic decat reducerea, asa ca factura a
fost actualizata ca fiind platita');

          UPDATE Facturi
          SET status = 'Platit'
          WHERE cod_factura = factura.cod_factura
          AND cod_adresa = factura.cod_adresa
          AND cod_client = factura.cod_client;

        WHEN FACTURA_DEJA_PLATITA THEN
          DBMS_OUTPUT.PUT_LINE('Factura a fost deja platita');
    END;
    DBMS_OUTPUT.PUT_LINE(' ');
  END LOOP;

  RETURN v_reducere_totala;
END;



PROCEDURE cerinta_9(v_nume Informatii_Clienti.nume%TYPE, v_prenume
Informatii_Clienti.prenume%TYPE) IS
```

```
CURSOR c_carduri (cod_client_param Clienti.cod_client%TYPE) IS
    SELECT DISTINCT cb.numar_card, cb.cod_securitate_card
    FROM Carduri_Bancare cb, Informatii_Bancare ib, Clienti c, Informatii_Clienti ic, Adrese a,
Facturi f
    WHERE cb.cod_client = cod_client_param
    AND ib.cod_client = cod_client_param
    AND ib.sold_curent > 2000
    AND a.cod_client = cod_client_param
    AND a.tara = 'Romania'
    AND f.cod_client = cod_client_param
    AND f.cod_adresa = a.cod_adresa
    AND f.status = 'Neplatit';

v_cod_client Clienti.cod_client%TYPE;
v_numar_card Carduri_bancare.numar_card%TYPE;
v_cod_securitate_card Carduri_bancare.cod_securitate_card%TYPE;
v_index_card NUMBER(4) := 1;
BEGIN

  SELECT c.cod_client      --aflare cod client
  INTO v_cod_client
  FROM Informatii_Clienti ic, Clienti c
  WHERE c.cod_informatii_client = ic.cod_informatii_client
  AND ic.nume = v_nume
  AND ic.prenume = v_prenume;

  OPEN c_carduri(v_cod_client);
  LOOP
    FETCH c_carduri INTO v_numar_card, v_cod_securitate_card;
    EXIT WHEN c_carduri%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Cardul ' || v_index_card || ': numar - ' || v_numar_card || ', cod
securitate - ' || v_cod_securitate_card);
    v_index_card := v_index_card + 1;
  END LOOP;

  UPDATE Facturi
  SET status = 'Platit'
  WHERE cod_client = v_cod_client;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE ('NO DATA FOUND!');
    WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE ('PREA MULTE
REZULTATE!');
    WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE ('ALTA EROARE!');
```

END;
END pachet_cerinta_13;

EXECUTE pachet_cerinta_13.cerinta_6;

```
--------------------------------------------------------------------------13--------------------------------------------------------------------------

CREATE OR REPLACE PACKAGE pachet_cerinta_13 AS
    PROCEDURE cerinta_6;
    PROCEDURE cerinta_7(v_numar_minim_plangeri IN NUMBER);
    FUNCTION cerinta_8(v_reducere IN NUMBER) RETURN NUMBER;
    PROCEDURE cerinta_9(v_nume Informatii_Clienti.nume%TYPE, v_prenume Informatii_Clienti.prenume%TYPE);
END pachet_cerinta_13;




CREATE OR REPLACE PACKAGE BODY pachet_cerinta_13 AS

PROCEDURE cerinta_6 IS
    TYPE vector_orase IS VARRAY(20) OF Adrese.oras%TYPE;
    TYPE tabel_orase IS TABLE OF Adrese.oras%TYPE INDEX BY PLS_INTEGER;
    TYPE tabel_clienti IS TABLE OF clienti.email%TYPE INDEX BY PLS_INTEGER;
    v_vector_orase vector_orase := vector_orase('Chisinau', 'Craiova', 'Bucuresti', 'Iasi', 'Bacau', 'Kiev', 'Beijing');
    v_tabel_orase tabel_orase;
    v_tabel_clienti tabel_clienti;     --retine id-ul pt un client
    v_numar_adrese NUMBER(4);
    v_minim_o_adresa BOOLEAN;
    v_email Clienti.email%TYPE;
    j NUMBER(4);
BEGIN

        SELECT DISTINCT oras          --punem toate orasele in v_tabel_orase
        BULK COLLECT INTO v_tabel_orase
        FROM Adrese;
```

Script Output | Query Result

Task completed in 0.034 seconds

```
Package PACHET_CERINTA_13 compiled


Package Body PACHET_CERINTA_13 compiled
```