SISTEMUL DE GESTIUNE A UNUI LANȚ DE MALL-URI

Mitu Iustin Aurelian 251

Ianuarie 2024

Contents

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

Modelul real va gestiona informații legate de funcționarea unui lanț internațional de mall-uri. Baza de date are scopul de a gestiona informațiile acestor mall-uri.

Acest lanț va cuprinde mai multe mall-uri din țări diferite. Fiecarui mall i se va ține evidența asupra numelui și dimensiunii acestuia (în metri pătrați). În interiorul fiecărui mall vor exista mai multe magazine. Magazinele vor emite un profit lunar (calculat în euro) și poate fi contactat prin numărul de telefon asociat.

Pentru a asigura paza și liniștea în interior, fiecare mall va avea un contract cu o unică firmă de securitate. Firma de securitate va conține angajați responsabili cu paza. Într-un magazin vor lucra mai mulți angajați. Deci, angajații din această bază de date pot fi doar de 3 tipuri, paznici (din cadrul firmei de securitate), interni (adică angajații fiecărui magazin în parte) și proprietari (adică proprietarul sau patronul unui magazin sau mai multe). Un intern poate lucra la mai multe magazine, iar într-un magazin pot lucra mai mulți angajați.

Entitatea produs reprezintă efectiv un singur produs. Produsul reprezintă bunul material pe care un client îl poate procura, sau chiar un serviciu de care clientul poate beneficia. Magazinele dețin mai multe produse. Dacă produsul este vândut, înseamnă că produsul respectiv va apărea într-o achiziție.

Clienții sunt persoane care vizitează mall-ul pentru a face achiziții. Totodată, aceștia pot efectua mai multe achiziții. Orice client are opțiunea de a face o reclamație, sau mai multe, unui magazin în cazul în care ceva nu pare la locul lui. Pentru a face o reclamație, clientul are nevoie de un motiv (pe care îl poate menționa în reclamație).

Un mall poate avea asociată o companie de promovare (sau mai multe), dar nu este obligatoriu. Această promovare are o dată când începe și o dată când se sfârșește. Promovarea are scopul de a face publicitate mall-ului cu telul de a atrage mai mulți clienți pentru a genera mai mult profit.

2. Realizați diagrama entitate-relație (ERD).

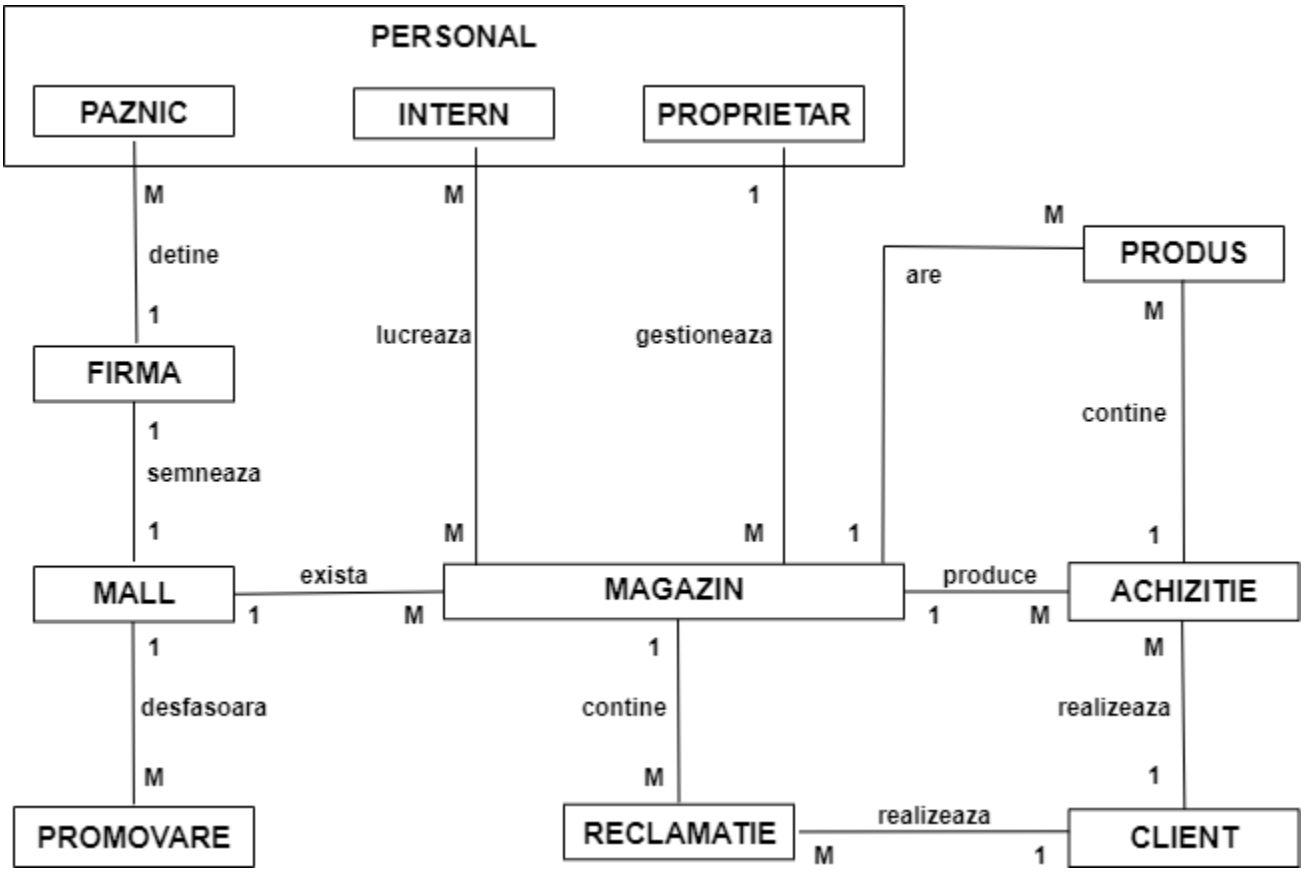
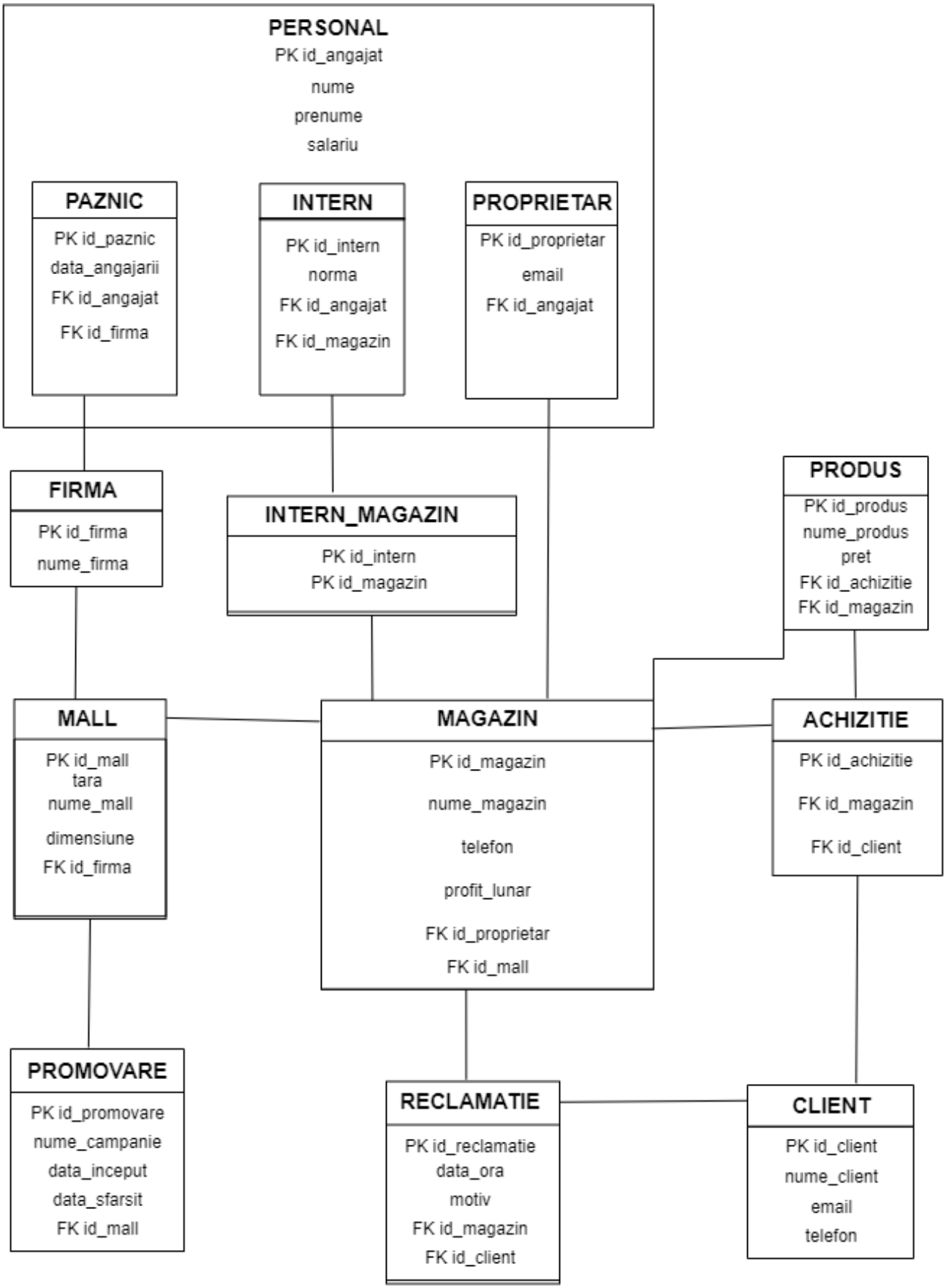


Figure 1: Diagrama entitate - relatie.

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


PERSONAL
PK id_angajat
nume
prenume
salariu
PAZNIC
PK id_paznic
data_angajarii
FK id_angajat
FK id_firma
INTERN
PK id_intern
norma
FK id_angajat
FK id_magazin
PROPRIETAR
PK id_proprietar
email
FK id_angajat
FIRMA
PK id_firma
nume_firma
INTERN_MAGAZIN
PK id_intern
PK id_magazin
PRODUS
PK id_produs
nume_produs
pret
FK id_achizitie
FK id_magazin
MALL
PK id_mall
tara
nume_mall
dimensiune
FK id_firma
MAGAZIN
PK id_magazin
nume_magazin
telefon
profit_lunar
FK id_proprietar
FK id_mall
ACHIZITIE
PK id_achizitie
FK id_magazin
FK id_client
PROMOVARE
PK id_promovare
nume_campanie
data_inceput
data_sfarsit
FK id_mall
RECLAMATIE
PK id_reclamatie
data_ora
motiv
FK id_magazin
FK id_client
CLIENT
PK id_client
nume_client
email
telefon

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

```
/*
DROP TABLE personal cascade constraints;
DROP TABLE intern cascade constraints;
DROP TABLE paznic cascade constraints;
DROP TABLE proprietar cascade constraints;
DROP TABLE intern_magazin cascade constraints;
DROP TABLE magazin cascade constraints;
DROP TABLE firma cascade constraints;
DROP TABLE mall cascade constraints;
DROP TABLE promovare cascade constraints;
DROP TABLE reclamatie cascade constraints;
DROP TABLE client cascade constraints;
DROP TABLE achizitie cascade constraints;
DROP TABLE produs cascade constraints;
*/
```

```
CREATE TABLE FIRMA (
    id_firma NUMBER(4) PRIMARY KEY,
    nume_firma VARCHAR2(50)
 );
CREATE TABLE MALL (
    id_mall NUMBER(4) PRIMARY KEY,
    tara VARCHAR2(50),
    nume_mall VARCHAR2(50),
    dimensiune NUMBER(7),
    id_firma NUMBER(4) NOT NULL,
    FOREIGN KEY (id_firma) REFERENCES FIRMA (id_firma)
);
 CREATE TABLE PROMOVARE (
    id_promovare NUMBER(4) PRIMARY KEY,
    nume_campanie VARCHAR2(50) ,
    data_inceput DATE ,
    data_sfarsit DATE ,
    id_mall NUMBER(4) ,
    FOREIGN KEY (id_mall) REFERENCES MALL (id_mall)
 );
 CREATE TABLE PERSONAL (
    id_angajat NUMBER(4) PRIMARY KEY,
```

```
    nume VARCHAR2(30),
    prenume VARCHAR2(30),
    salariu NUMBER(4)
 );
 CREATE TABLE PROPRIETAR (
    id_proprietar NUMBER(4) PRIMARY KEY,
    email VARCHAR2(30) ,
    id_angajat NUMBER(4),
    FOREIGN KEY (id_angajat) REFERENCES PERSONAL (id_angajat)
);
 CREATE TABLE PAZNIC (
    id_paznic NUMBER(4) PRIMARY KEY,
    data_angajarii DATE,
    id_angajat NUMBER(4),
    id_firma NUMBER(4),
    FOREIGN KEY (id_angajat) REFERENCES PERSONAL (id_angajat),
    FOREIGN KEY (id_firma) REFERENCES FIRMA (id_firma)
);
CREATE TABLE MAGAZIN (
   id_magazin NUMBER(4) PRIMARY KEY,
   nume_magazin VARCHAR2(30),
   telefon VARCHAR2(20),
```

```
   profit_lunar NUMBER(4),
   id_proprietar NUMBER(4),
   id_mall NUMBER(4),
   FOREIGN KEY (id_proprietar) REFERENCES PROPRIETAR (id_proprietar),
   FOREIGN KEY (id_mall) REFERENCES MALL (id_mall)
);
 CREATE TABLE INTERN (
    id_intern NUMBER(4) PRIMARY KEY,
    norma VARCHAR2(30),
    id_angajat NUMBER(4),
    FOREIGN KEY (id_angajat) REFERENCES PERSONAL (id_angajat)
);
CREATE TABLE INTERN_MAGAZIN (
   id_intern NUMBER(4),
   id_magazin NUMBER(4),
   PRIMARY KEY (id_intern, id_magazin),
   FOREIGN KEY (id_intern) REFERENCES INTERN(id_intern),
   FOREIGN KEY (id_magazin) REFERENCES MAGAZIN(id_magazin)
);
CREATE TABLE CLIENT (
   id_client NUMBER(4) PRIMARY KEY,
   nume_client VARCHAR2(30),
```

```
   email VARCHAR2(30),
   telefon VARCHAR2(30)
);
CREATE TABLE RECLAMATIE (
   id_reclamatie NUMBER(4) PRIMARY KEY,
   data_ora TIMESTAMP,
   motiv VARCHAR2(1000),
   id_magazin NUMBER(4),
   id_client NUMBER(4),
   FOREIGN KEY (id_magazin) REFERENCES MAGAZIN (id_magazin),
   FOREIGN KEY (id_client) REFERENCES CLIENT (id_client)
);
CREATE TABLE ACHIZITIE (
   id_achizitie NUMBER(4) PRIMARY KEY,
   id_magazin NUMBER(4),
   id_client NUMBER(4),
   FOREIGN KEY (id_magazin) REFERENCES MAGAZIN (id_magazin),
   FOREIGN KEY (id_client) REFERENCES CLIENT (id_client)
);
CREATE TABLE PRODUS (
   id_produs NUMBER(4) PRIMARY KEY,
   nume_produs VARCHAR2(30),
```

```
    pret NUMBER(4),
    id_magazin NUMBER(4),
    id_achizitie NUMBER(4),
    FOREIGN KEY (id_magazin) REFERENCES MAGAZIN (id_magazin),
    FOREIGN KEY (id_achizitie) REFERENCES ACHIZITIE (id_achizitie)
);
```

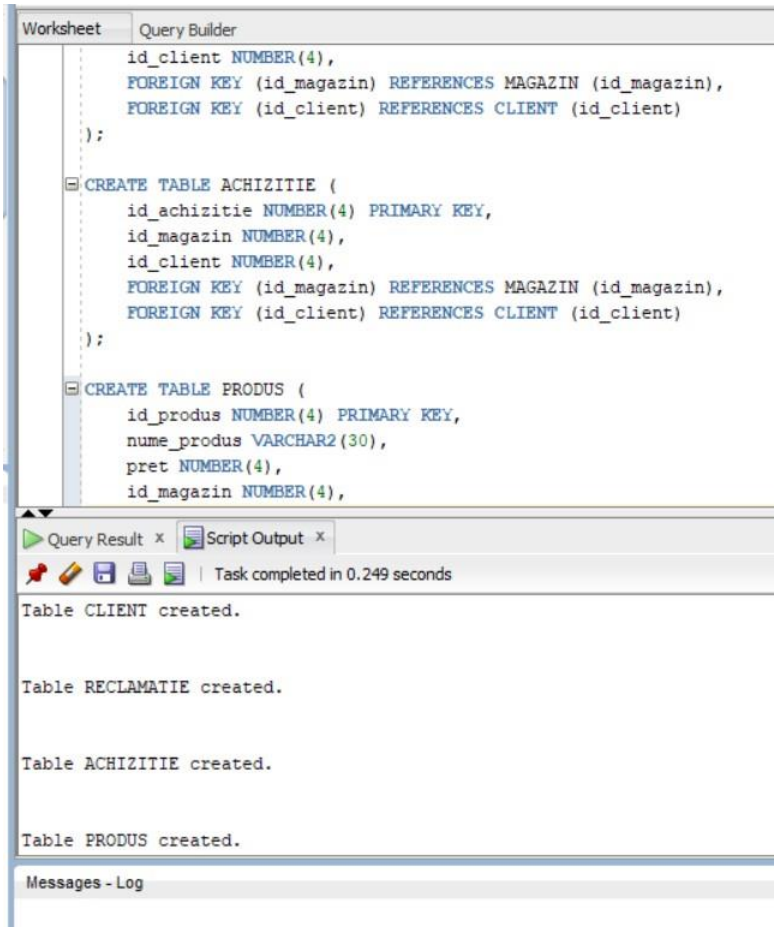


Figure 3: EX 4.

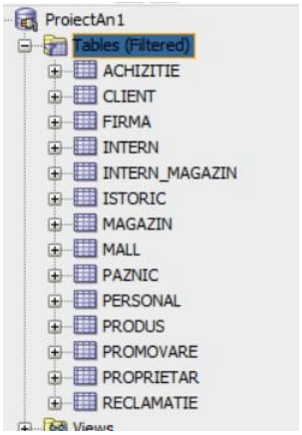

ProiectAn1
Tables (Filtered)
ACHIZITIE
CLIENT
FIRMA
INTERN
INTERN_MAGAZIN
ISTORIC
MAGAZIN
MALL
PAZNIC
PERSONAL
PRODUS
PROMOVARE
PROPRIETAR
RECLAMATIE

Figure 4: EX 4.

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

```
-- FIRMA


INSERT INTO FIRMA VALUES (1, 'SecurGuard Solutions');

INSERT INTO FIRMA VALUES (2, 'SafeWatch Services');

INSERT INTO FIRMA VALUES (3, 'ShieldSecurity Group');

INSERT INTO FIRMA VALUES (4, 'GuardianForce Systems');

INSERT INTO FIRMA VALUES (5, 'SecurePath Inc.');

INSERT INTO FIRMA VALUES (6, 'VigilantWatch Solutions');

INSERT INTO FIRMA VALUES (7, 'SecureShield Corp.');

INSERT INTO FIRMA VALUES (8, 'Safeguard Masters');

INSERT INTO FIRMA VALUES (9, 'EagleEye Security');

INSERT INTO FIRMA VALUES (10, 'FortressGuard Systems');



-- MALL
```

```
INSERT INTO MALL VALUES (1, 'Romania', 'Mega Mall', 150000, 1);
INSERT INTO MALL VALUES (2, 'Romania', 'City Mall', 120000, 2);
INSERT INTO MALL VALUES (3, 'Spania', 'Barcelona Shopping Center', 180000, 3);
INSERT INTO MALL VALUES (4, 'Franta', 'Galerie Lafayette', 130000, 4);
INSERT INTO MALL VALUES (5, 'Italia', 'Milano Galleria', 110000, 5);
INSERT INTO MALL VALUES (6, 'Germania', 'Berlin Fashion Square', 160000, 6);
INSERT INTO MALL VALUES (7, 'Olanda', 'Amsterdam Retail Plaza', 90000, 7);
INSERT INTO MALL VALUES (8, 'Suedia', 'Stockholm City Center', 140000, 8);
INSERT INTO MALL VALUES (9, 'Elvetia', 'Zurich Luxury Mall', 100000, 9);
INSERT INTO MALL VALUES (10, 'Austria', 'Vienna Shopping Paradise', 110000, 10);



-- PERSONAL


INSERT INTO PERSONAL VALUES (1, 'Ana', 'Popescu', 3500);
INSERT INTO PERSONAL VALUES (2, 'Radu', 'Ionescu', 4000);
INSERT INTO PERSONAL VALUES (3, 'Elena', 'Dumitru', 3800);
INSERT INTO PERSONAL VALUES (4, 'Vlad', 'Georgescu', 4200);
INSERT INTO PERSONAL VALUES (5, 'Ioana', 'Stoica', 3600);
INSERT INTO PERSONAL VALUES (6, 'Mihai', 'Petrov', 3800);
INSERT INTO PERSONAL VALUES (7, 'Andreea', 'Vasilescu', 4200);
```

```
INSERT INTO PERSONAL VALUES (8, 'Gabriel', 'Marin', 3500);

INSERT INTO PERSONAL VALUES (9, 'Cristina', 'Neagu', 4000);

INSERT INTO PERSONAL VALUES (10, 'Alex', 'Stancu', 3600);

INSERT INTO PERSONAL VALUES (11, 'Laura', 'Constantin', 3800);

INSERT INTO PERSONAL VALUES (12, 'Bogdan', 'Alexandrescu', 4200);

INSERT INTO PERSONAL VALUES (13, 'Anda', 'Mihai', 3500);

INSERT INTO PERSONAL VALUES (14, 'Razvan', 'Florescu', 4000);

INSERT INTO PERSONAL VALUES (15, 'Simona', 'Nistor', 3600);

INSERT INTO PERSONAL VALUES (16, 'Catalin', 'Barbu', 3900);

INSERT INTO PERSONAL VALUES (17, 'Alina', 'Dumitrache', 4100);

INSERT INTO PERSONAL VALUES (18, 'Ionut', 'Popa', 3700);

INSERT INTO PERSONAL VALUES (19, 'Mihaela', 'Dinu', 4300);

INSERT INTO PERSONAL VALUES (20, 'Sorin', 'Munteanu', 3800);

INSERT INTO PERSONAL VALUES (21, 'Ana-Maria', 'Ionescu', 4200);

INSERT INTO PERSONAL VALUES (22, 'Victor', 'Vasile', 3500);

INSERT INTO PERSONAL VALUES (23, 'Cristian', 'Stoian', 4000);

INSERT INTO PERSONAL VALUES (24, 'Elena', 'Popa', 3600);

INSERT INTO PERSONAL VALUES (25, 'Radu', 'Iancu', 3900);

INSERT INTO PERSONAL VALUES (26, 'Gabriela', 'Gheorghe', 4100);

INSERT INTO PERSONAL VALUES (27, 'Valentin', 'Grigorescu', 3700);

INSERT INTO PERSONAL VALUES (28, 'Adela', 'Dumitrescu', 4300);

INSERT INTO PERSONAL VALUES (29, 'Dorin', 'Moldovan', 3800);
```

```
INSERT INTO PERSONAL VALUES (30, 'Silvia', 'Istrate', 4200);




-- PAZNIC


INSERT INTO PAZNIC VALUES (1, TO_DATE('2023-01-15', 'YYYY-MM-DD'), 1, 1);

INSERT INTO PAZNIC VALUES (2, TO_DATE('2023-02-20', 'YYYY-MM-DD'), 2, 2);

INSERT INTO PAZNIC VALUES (3, TO_DATE('2023-03-25', 'YYYY-MM-DD'), 3, 3);

INSERT INTO PAZNIC VALUES (4, TO_DATE('2023-04-10', 'YYYY-MM-DD'), 4, 4);

INSERT INTO PAZNIC VALUES (5, TO_DATE('2023-05-12', 'YYYY-MM-DD'), 5, 5);

INSERT INTO PAZNIC VALUES (6, TO_DATE('2023-06-18', 'YYYY-MM-DD'), 6, 6);

INSERT INTO PAZNIC VALUES (7, TO_DATE('2023-07-22', 'YYYY-MM-DD'), 7, 7);

INSERT INTO PAZNIC VALUES (8, TO_DATE('2023-08-30', 'YYYY-MM-DD'), 8, 8);

INSERT INTO PAZNIC VALUES (9, TO_DATE('2023-09-05', 'YYYY-MM-DD'), 9, 9);

INSERT INTO PAZNIC VALUES (10, TO_DATE('2023-10-10', 'YYYY-MM-DD'), 10, 10);




-- INTERN


INSERT INTO INTERN VALUES (1, 'Full-time', 11);

INSERT INTO INTERN VALUES (2, 'Part-time', 12);
```

```
INSERT INTO INTERN VALUES (3, 'Full-time', 13);

INSERT INTO INTERN VALUES (4, 'Part-time', 14);

INSERT INTO INTERN VALUES (5, 'Full-time', 15);

INSERT INTO INTERN VALUES (6, 'Part-time', 16);

INSERT INTO INTERN VALUES (7, 'Full-time', 17);

INSERT INTO INTERN VALUES (8, 'Part-time', 18);

INSERT INTO INTERN VALUES (9, 'Full-time', 19);

INSERT INTO INTERN VALUES (10, 'Part-time', 20);

INSERT INTO INTERN VALUES (11, 'Part-time', 21);

INSERT INTO INTERN VALUES (12, 'Full-time', 22);

INSERT INTO INTERN VALUES (13, 'Part-time', 23);

INSERT INTO INTERN VALUES (14, 'Full-time', 24);

INSERT INTO INTERN VALUES (15, 'Part-time', 25);



-- PROPRIETAR


INSERT INTO PROPRIETAR VALUES (1, 'gabriela@email.com', 26);

INSERT INTO PROPRIETAR VALUES (2, 'valentin@email.com', 27);

INSERT INTO PROPRIETAR VALUES (3, 'adela@email.com', 28);

INSERT INTO PROPRIETAR VALUES (4, 'dorin@email.com', 29);

INSERT INTO PROPRIETAR VALUES (5, 'silvia@email.com', 30);
```

```
-- MAGAZIN



INSERT INTO MAGAZIN VALUES (1, 'Electronics Emporium', '1234567890', 5000, 1, 1);
INSERT INTO MAGAZIN VALUES (2, 'Fashion Boutique', '9876543210', 6000, 2, 2);
INSERT INTO MAGAZIN VALUES (3, 'Home Decor Haven', '1231231234', 4500, 3, 3);
INSERT INTO MAGAZIN VALUES (4, 'Beauty Corner', '4567890123', 5500, 4, 4);
INSERT INTO MAGAZIN VALUES (5, 'Tech World', '7890123456', 7000, 5, 5);
INSERT INTO MAGAZIN VALUES (6, 'Sporty Gear', '2345678901', 4800, 1, 6);
INSERT INTO MAGAZIN VALUES (7, 'Kitchen Essentials', '8901234567', 5200, 2, 7);
INSERT INTO MAGAZIN VALUES (8, 'Book Nook', '3456789012', 5800, 3, 8);
INSERT INTO MAGAZIN VALUES (9, 'Gourmet Delights', '6789012345', 4900, 4, 9);
INSERT INTO MAGAZIN VALUES (10, 'Outdoor Adventures', '9012345678', 6500, 5, 10);
INSERT INTO MAGAZIN VALUES (11, 'Tech Haven', '3456789123', 5500, 1, 1);
INSERT INTO MAGAZIN VALUES (12, 'Chic Couture', '7890123456', 6000, 2, 2);
INSERT INTO MAGAZIN VALUES (13, 'Living Luxe', '2345678901', 4800, 3, 3);
INSERT INTO MAGAZIN VALUES (14, 'Health Hub', '9012345678', 5200, 4, 4);
INSERT INTO MAGAZIN VALUES (15, 'Music Melody', '3456789123', 5800, 5, 5);
INSERT INTO MAGAZIN VALUES (16, 'Fitness Fanatic', '4567890123', 4900, 1, 6);
INSERT INTO MAGAZIN VALUES (17, 'Culinary Creations', '7890123456', 7000, 2, 7);
```

```
INSERT INTO MAGAZIN VALUES (18, 'Artistic Attire', '1231231234', 4500, 3, 8);

INSERT INTO MAGAZIN VALUES (19, 'Pet Paradise', '8901234567', 5500, 4, 9);

INSERT INTO MAGAZIN VALUES (20, 'Travel Treasures', '2345678901', 6500, 5, 10);




-- PROMOVARE



INSERT INTO PROMOVARE VALUES (1, 'Sale Extravaganza', TO_DATE('2023-02-15', 'YYYY-MM-DD'),
TO_DATE('2023-02-28', 'YYYY-MM-DD'), 1);

INSERT INTO PROMOVARE VALUES (2, 'Fashion Frenzy', TO_DATE('2023-03-10', 'YYYY-MM-DD'),
TO_DATE('2023-03-20', 'YYYY-MM-DD'), 2);

INSERT INTO PROMOVARE VALUES (3, 'Home Decor Delights', TO_DATE('2023-04-05', 'YYYY-MM-DD'),
TO_DATE('2023-04-15', 'YYYY-MM-DD'), 3);

INSERT INTO PROMOVARE VALUES (4, 'Beauty Bliss', TO_DATE('2023-05-01', 'YYYY-MM-DD'),
TO_DATE('2023-05-10', 'YYYY-MM-DD'), 4);

INSERT INTO PROMOVARE VALUES (5, 'Tech Spectacle', TO_DATE('2023-06-15', 'YYYY-MM-DD'),
TO_DATE('2023-06-30', 'YYYY-MM-DD'), 5);

INSERT INTO PROMOVARE VALUES (6, 'Sporty Savings', TO_DATE('2023-07-10', 'YYYY-MM-DD'),
TO_DATE('2023-07-20', 'YYYY-MM-DD'), 6);

INSERT INTO PROMOVARE VALUES (7, 'Kitchen Carnival', TO_DATE('2023-08-05', 'YYYY-MM-DD'),
TO_DATE('2023-08-15', 'YYYY-MM-DD'), 7);

INSERT INTO PROMOVARE VALUES (8, 'Book Bonanza', TO_DATE('2023-09-01', 'YYYY-MM-DD'),
TO_DATE('2023-09-10', 'YYYY-MM-DD'), 8);

INSERT INTO PROMOVARE VALUES (9, 'Gourmet Gala', TO_DATE('2023-10-15', 'YYYY-MM-DD'),
TO_DATE('2023-10-30', 'YYYY-MM-DD'), 9);

INSERT INTO PROMOVARE VALUES (10, 'Outdoor Oasis', TO_DATE('2023-11-10', 'YYYY-MM-DD'),
TO_DATE('2023-11-20', 'YYYY-MM-DD'), 10);
```

```
-- CLIENT


INSERT INTO CLIENT VALUES (1, 'Ana Maria', 'ana.maria@email.com', '1234567890');
INSERT INTO CLIENT VALUES (2, 'George Popescu', 'george.popescu@email.com', '9876543210');
INSERT INTO CLIENT VALUES (3, 'Elena Ionescu', 'elena.ionescu@email.com', '5678901234');
INSERT INTO CLIENT VALUES (4, 'Alexandru Vlad', 'alexandru.vlad@email.com', '6789012345');
INSERT INTO CLIENT VALUES (5, 'Andreea Stoica', 'andreea.stoica@email.com', '3456789012');
INSERT INTO CLIENT VALUES (6, 'Cristian Dumitru', 'cristian.dumitru@email.com', '8901234567');
INSERT INTO CLIENT VALUES (7, 'Diana Georgescu', 'diana.georgescu@email.com', '4567890123');
INSERT INTO CLIENT VALUES (8, 'Vlad Gheorghe', 'vlad.gheorghe@email.com', '2345678901');
INSERT INTO CLIENT VALUES (9, 'Roxana Moldovan', 'roxana.moldovan@email.com', '7890123456');
INSERT INTO CLIENT VALUES (10, 'Mihai Radu', 'mihai.radu@email.com', '9012345678');
INSERT INTO CLIENT VALUES (11, 'Alina Popa', 'alina.popa@email.com', '6789123456');
INSERT INTO CLIENT VALUES (12, 'Ionut Dima', 'ionut.dima@email.com', '7891234567');
INSERT INTO CLIENT VALUES (13, 'Simona Tudor', 'simona.tudor@email.com', '8902345678');
INSERT INTO CLIENT VALUES (14, 'Victor Stanescu', 'victor.stanescu@email.com', '2345678901');
INSERT INTO CLIENT VALUES (15, 'Eva Munteanu', 'eva.munteanu@email.com', '1234567890');
INSERT INTO CLIENT VALUES (16, 'Robert Florescu', 'robert.florescu@email.com', '7890123456');
INSERT INTO CLIENT VALUES (17, 'Ana Dragomir', 'ana.dragomir@email.com', '2345678901');
```

```
INSERT INTO CLIENT VALUES (18, 'Catalin Vasile', 'catalin.vasile@email.com', '5678901234');

INSERT INTO CLIENT VALUES (19, 'Anda Marinescu', 'anda.marinescu@email.com', '6789012345');

INSERT INTO CLIENT VALUES (20, 'Adrian Radulescu', 'adrian.radulescu@email.com', '8901234567');

INSERT INTO CLIENT VALUES (21, 'Maria Alexandrescu', 'maria.alexandrescu@email.com',
'02394325432');



-- RECLAMATIE




INSERT INTO RECLAMATIE VALUES (1, TO_TIMESTAMP('2023-02-15 08:30:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Produs defect', 1, 1);

INSERT INTO RECLAMATIE VALUES (2, TO_TIMESTAMP('2023-03-10 12:45:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Servicii proaste', 2, 2);

INSERT INTO RECLAMATIE VALUES (3, TO_TIMESTAMP('2023-04-05 14:20:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Experienta dezamagitoare', 3, 3);

INSERT INTO RECLAMATIE VALUES (4, TO_TIMESTAMP('2023-05-01 10:10:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Probleme de facturare', 4, 4);

INSERT INTO RECLAMATIE VALUES (5, TO_TIMESTAMP('2023-06-15 16:30:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Livrare intarziata', 5, 5);

INSERT INTO RECLAMATIE VALUES (6, TO_TIMESTAMP('2023-07-10 09:15:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Produs neconform', 6, 6);

INSERT INTO RECLAMATIE VALUES (7, TO_TIMESTAMP('2023-08-05 13:50:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Personal nepoliticos', 7, 7);

INSERT INTO RECLAMATIE VALUES (8, TO_TIMESTAMP('2023-09-01 11:05:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Probleme tehnice', 8, 8);

INSERT INTO RECLAMATIE VALUES (9, TO_TIMESTAMP('2023-10-15 15:40:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Produs deteriorat', 9, 9);
```

```
INSERT INTO RECLAMATIE VALUES (10, TO_TIMESTAMP('2023-11-10 17:20:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Servicii client slabe', 10, 10);

INSERT INTO RECLAMATIE VALUES (11, TO_TIMESTAMP('2024-01-05 09:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Probleme de stoc', 3, 11);

INSERT INTO RECLAMATIE VALUES (12, TO_TIMESTAMP('2024-01-20 14:30:00', 'YYYY-MM-DD
HH24:MI:SS'), 'Retur refuzat', 3, 12);




-- ACHIZITIE




INSERT INTO ACHIZITIE VALUES (1, 1, 1);

INSERT INTO ACHIZITIE VALUES (2, 2, 2);

INSERT INTO ACHIZITIE VALUES (3, 3, 3);

INSERT INTO ACHIZITIE VALUES (4, 4, 4);

INSERT INTO ACHIZITIE VALUES (5, 5, 5);

INSERT INTO ACHIZITIE VALUES (6, 6, 6);

INSERT INTO ACHIZITIE VALUES (7, 7, 7);

INSERT INTO ACHIZITIE VALUES (8, 8, 8);

INSERT INTO ACHIZITIE VALUES (9, 9, 9);

INSERT INTO ACHIZITIE VALUES (10, 10, 10);

INSERT INTO ACHIZITIE VALUES (11, 1, 11);

INSERT INTO ACHIZITIE VALUES (12, 2, 12);
```

```
INSERT INTO ACHIZITIE VALUES (13, 3, 13);

INSERT INTO ACHIZITIE VALUES (14, 4, 14);

INSERT INTO ACHIZITIE VALUES (15, 5, 15);

INSERT INTO ACHIZITIE VALUES (16, 6, 16);

INSERT INTO ACHIZITIE VALUES (17, 7, 17);

INSERT INTO ACHIZITIE VALUES (18, 8, 18);

INSERT INTO ACHIZITIE VALUES (19, 9, 19);

INSERT INTO ACHIZITIE VALUES (20, 10, 20);

INSERT INTO ACHIZITIE VALUES (21, 1, 1);

INSERT INTO ACHIZITIE VALUES (22, 2, 2);

INSERT INTO ACHIZITIE VALUES (23, 3, 3);

INSERT INTO ACHIZITIE VALUES (24, 4, 4);

INSERT INTO ACHIZITIE VALUES (25, 5, 5);

INSERT INTO ACHIZITIE VALUES (26, 6, 6);

INSERT INTO ACHIZITIE VALUES (27, 7, 7);

INSERT INTO ACHIZITIE VALUES (28, 8, 8);

INSERT INTO ACHIZITIE VALUES (29, 9, 9);

INSERT INTO ACHIZITIE VALUES (30, 10, 10);

INSERT INTO ACHIZITIE VALUES (31, 10, 1);



-- PRODUS
```

```
INSERT INTO PRODUS VALUES (1, 'Laptop', 1500, 1, 1);
INSERT INTO PRODUS VALUES (2, 'Smartphone', 800, 2, 2);
INSERT INTO PRODUS VALUES (3, 'TV LED', 1200, 3, 3);
INSERT INTO PRODUS VALUES (4, 'Cafetiera', 100, 4, 4);
INSERT INTO PRODUS VALUES (5, 'Tracker Fitness', 50, 5, 5);
INSERT INTO PRODUS VALUES (6, 'Boxa Bluetooth', 70, 6, 6);
INSERT INTO PRODUS VALUES (7, 'Mouse Gaming', 60, 7, 7);
INSERT INTO PRODUS VALUES (8, 'Aparat Foto DSLR', 800, 8, 8);
INSERT INTO PRODUS VALUES (9, 'Hard Disk Extern', 120, 9, 9);
INSERT INTO PRODUS VALUES (10, 'Periuta de Dinti Electrica', 30, 10, 10);
INSERT INTO PRODUS VALUES (11, 'Casti', 50, 1, 11);
INSERT INTO PRODUS VALUES (12, 'Tableta', 400, 2, 12);
INSERT INTO PRODUS VALUES (13, 'Rasnita de Cafea', 40, 3, 13);
INSERT INTO PRODUS VALUES (14, 'Incarcator Portabil', 20, 4, 14);
INSERT INTO PRODUS VALUES (15, 'Ceas Smart', 100, 5, 15);
INSERT INTO PRODUS VALUES (16, 'Tastatura Wireless', 30, 6, 16);
INSERT INTO PRODUS VALUES (17, 'Casti Bluetooth', 40, 7, 17);
INSERT INTO PRODUS VALUES (18, 'Drona', 500, 8, 18);
INSERT INTO PRODUS VALUES (19, 'Set Casti Bluetooth', 60, 9, 19);
INSERT INTO PRODUS VALUES (20, 'Imprimanta', 150, 10, 20);
```

```
INSERT INTO PRODUS VALUES (21, 'Lampa Birou', 25, 1, NULL);
INSERT INTO PRODUS VALUES (22, 'Mouse Wireless', 20, 2, NULL);
INSERT INTO PRODUS VALUES (23, 'Becuri Smart', 35, 3, NULL);
INSERT INTO PRODUS VALUES (24, 'SSD Extern', 80, 4, NULL);
INSERT INTO PRODUS VALUES (25, 'Banda Fitness', 45, 5, NULL);
INSERT INTO PRODUS VALUES (26, 'Priza Multiplicatoare', 15, 6, NULL);
INSERT INTO PRODUS VALUES (27, 'Termostat Smart', 70, 7, NULL);
INSERT INTO PRODUS VALUES (28, 'Tastatura Gaming', 90, 8, NULL);
INSERT INTO PRODUS VALUES (29, 'Webcam', 50, 9, NULL);
INSERT INTO PRODUS VALUES (30, 'Incarcator Wireless', 25, 10, NULL);
INSERT INTO PRODUS VALUES (31, 'SSD Portabil', 120, 1, NULL);
INSERT INTO PRODUS VALUES (32, 'Cantar Smart', 40, 2, NULL);
INSERT INTO PRODUS VALUES (33, 'E-reader', 60, 3, NULL);
INSERT INTO PRODUS VALUES (34, 'Camera Actiune', 100, 4, NULL);
INSERT INTO PRODUS VALUES (35, 'Aspirator Robot', 200, 5, NULL);
INSERT INTO PRODUS VALUES (36, 'Usa Smart', 80, 6, NULL);
INSERT INTO PRODUS VALUES (37, 'Camera Auto', 50, 7, NULL);
INSERT INTO PRODUS VALUES (38, 'Router Wireless', 70, 8, NULL);
INSERT INTO PRODUS VALUES (39, 'Proiector Portabil', 120, 9, NULL);
INSERT INTO PRODUS VALUES (40, 'Casti Wireless', 30, 10, NULL);
INSERT INTO PRODUS VALUES (41, 'Camera Instant', 60, 1, NULL);
INSERT INTO PRODUS VALUES (42, 'Purificator Aer', 150, 2, NULL);
```

```
INSERT INTO PRODUS VALUES (43, 'Oglinda Smart', 200, 3, NULL);

INSERT INTO PRODUS VALUES (44, 'Frigider Mini', 80, 4, NULL);

INSERT INTO PRODUS VALUES (45, 'Termometru Bluetooth', 35, 5, NULL);

INSERT INTO PRODUS VALUES (46, 'Prize Smart', 25, 6, NULL);

INSERT INTO PRODUS VALUES (47, 'Camera Supraveghere', 70, 7, NULL);

INSERT INTO PRODUS VALUES (48, 'Yala Smart', 120, 8, NULL);

INSERT INTO PRODUS VALUES (49, 'Proiector Mini', 50, 9, NULL);

INSERT INTO PRODUS VALUES (50, 'Frigider Smart', 500, 10, NULL);

INSERT INTO PRODUS VALUES (51, 'Scrumiera', 30, 5, 31);

INSERT INTO PRODUS VALUES (52, 'Pat', 5000, 9, 30);




-- INTERN_MAGAZIN


INSERT INTO INTERN_MAGAZIN VALUES (1, 1);

INSERT INTO INTERN_MAGAZIN VALUES (2, 2);

INSERT INTO INTERN_MAGAZIN VALUES (3, 3);

INSERT INTO INTERN_MAGAZIN VALUES (4, 4);

INSERT INTO INTERN_MAGAZIN VALUES (5, 5);

INSERT INTO INTERN_MAGAZIN VALUES (6, 6);

INSERT INTO INTERN_MAGAZIN VALUES (7, 7);
```

```
INSERT INTO INTERN_MAGAZIN VALUES (8, 8);

INSERT INTO INTERN_MAGAZIN VALUES (9, 9);

INSERT INTO INTERN_MAGAZIN VALUES (10, 10);

INSERT INTO INTERN_MAGAZIN VALUES (11, 11);

INSERT INTO INTERN_MAGAZIN VALUES (12, 12);

INSERT INTO INTERN_MAGAZIN VALUES (13, 13);

INSERT INTO INTERN_MAGAZIN VALUES (14, 14);

INSERT INTO INTERN_MAGAZIN VALUES (15, 15);

INSERT INTO INTERN_MAGAZIN VALUES (1, 16);

INSERT INTO INTERN_MAGAZIN VALUES (2, 17);

INSERT INTO INTERN_MAGAZIN VALUES (3, 18);

INSERT INTO INTERN_MAGAZIN VALUES (4, 19);

INSERT INTO INTERN_MAGAZIN VALUES (5, 20);

INSERT INTO INTERN_MAGAZIN VALUES (6, 1);

INSERT INTO INTERN_MAGAZIN VALUES (7, 2);

INSERT INTO INTERN_MAGAZIN VALUES (8, 3);

INSERT INTO INTERN_MAGAZIN VALUES (9, 4);

INSERT INTO INTERN_MAGAZIN VALUES (10, 5);

INSERT INTO INTERN_MAGAZIN VALUES (11, 6);

INSERT INTO INTERN_MAGAZIN VALUES (12, 7);

INSERT INTO INTERN_MAGAZIN VALUES (13, 8);

INSERT INTO INTERN_MAGAZIN VALUES (14, 9);
```

```
INSERT INTO INTERN_MAGAZIN VALUES (15, 10);

INSERT INTO INTERN_MAGAZIN VALUES (1, 13);

INSERT INTO INTERN_MAGAZIN VALUES (2, 14);
```

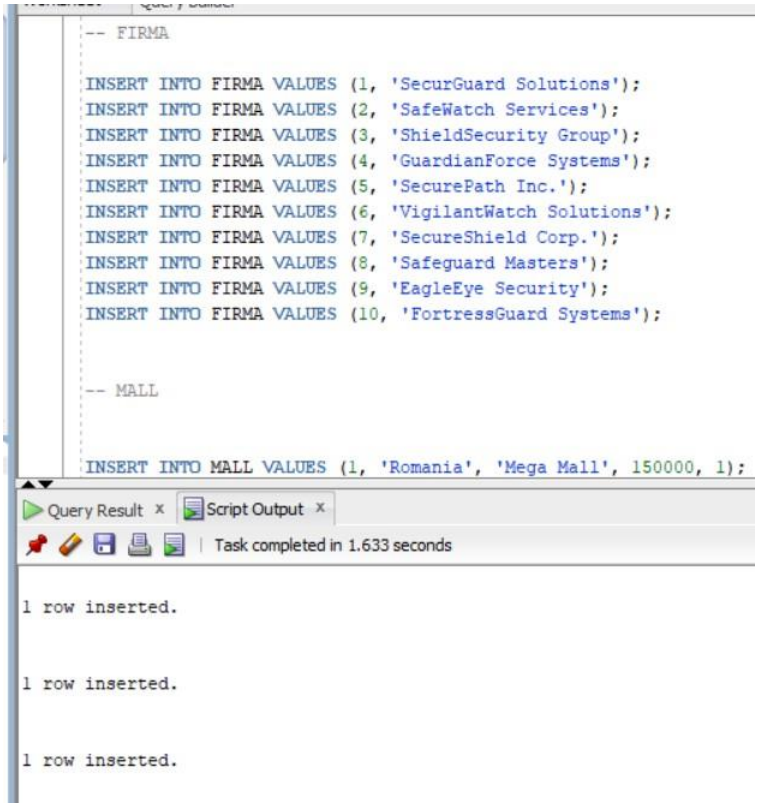
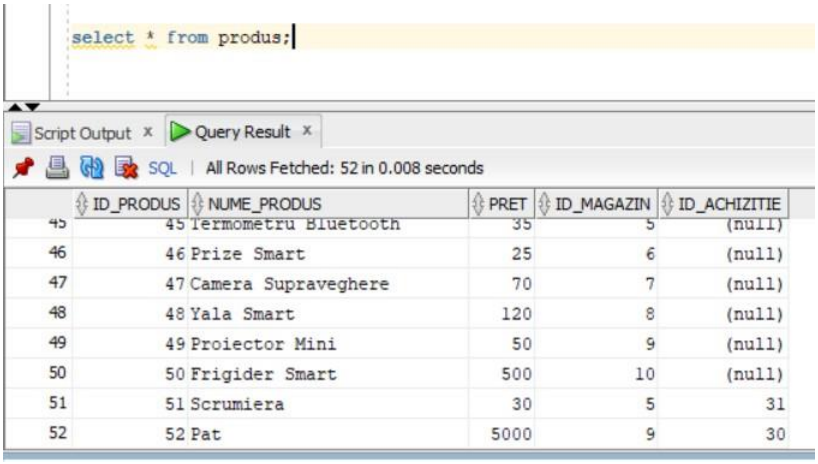


Figure 5: EX 5.



	ID_PRODUS	NUME_PRODUS	PRET	ID_MAGAZIN	ID_ACHIZITIE
45	45	Termometru Bluetooth	35	5	(null)
46	46	Prize Smart	25	6	(null)
47	47	Camera Supraveghere	70	7	(null)
48	48	Yala Smart	120	8	(null)
49	49	Proiector Mini	50	9	(null)
50	50	Frigider Smart	500	10	(null)
51	51	Scrumiera	30	5	31
52	52	Pat	5000	9	30

Figure 6: EX 5.

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

```
/*
Sa se afiseze lista de cumparaturi a tuturor clientilor.
*/


CREATE OR REPLACE PROCEDURE ex6 IS

   TYPE tablou_indexat IS TABLE OF CLIENT.id_client%TYPE INDEX BY PLS_INTEGER;

   TYPE tablou_imbricat IS TABLE OF VARCHAR2(30);

   TYPE vector IS VARRAY(10) OF NUMBER;



   id_clienti tablou_indexat;

   id_achizitii tablou_indexat;

   produse_cumparate tablou_imbricat := tablou_imbricat();

   preturi vector := vector();



   nr_produse NUMBER(4);

BEGIN

   -- obtinem id-urile clientilor

   SELECT id_client BULK COLLECT INTO id_clienti FROM CLIENT;
```

```
-- nr de produse in total
SELECT COUNT(*) INTO nr_produse FROM produs;


-- parcurgem toti clientii
FOR i IN 1..id_clienti.COUNT
LOOP


   -- pentru a obtine id-urile achizitiilor clientului
   SELECT id_achizitie
   BULK COLLECT INTO id_achizitii
   FROM achizitie
   WHERE id_client = id_clienti(i);


   -- pentru fiecare achizitie in care a participat clientul, se vor
   -- parcurge toate produsele pentru a le gasi pe cele care au
   -- fost cumparate in achizitia respectiva
   FOR j IN 1..id_achizitii.COUNT
   LOOP
      FOR achizitie_rec IN (SELECT nume_produs, pret, id_produs
                  FROM produs
                  WHERE id_achizitie = id_achizitii(j))
```

```
    LOOP
      produse_cumparate.EXTEND;
      produse_cumparate(produse_cumparate.LAST) := achizitie_rec.nume_produs;

      preturi.EXTEND;
      preturi(preturi.LAST) := achizitie_rec.pret;
    END LOOP;
  END LOOP;

  -- afisare pt fiecare produs
  DBMS_OUTPUT.PUT_LINE('ID Client: ' || id_clienti(i));

  FOR j IN 1..produse_cumparate.COUNT
  LOOP
    DBMS_OUTPUT.PUT_LINE('  Produs: ' || produse_cumparate(j) || ', Pret: ' || preturi(j));
  END LOOP;

  -- cazul in care un client nu a cumparat niciun produs
  IF produse_cumparate.COUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('  Nicio achizitie.');
  END IF;
```

```
    -- stergerea elementelor din tablouri
    produse_cumparate.delete;
    preturi := vector();
  END LOOP;
END ex6;
/

BEGIN
  ex6;
END;
/
```

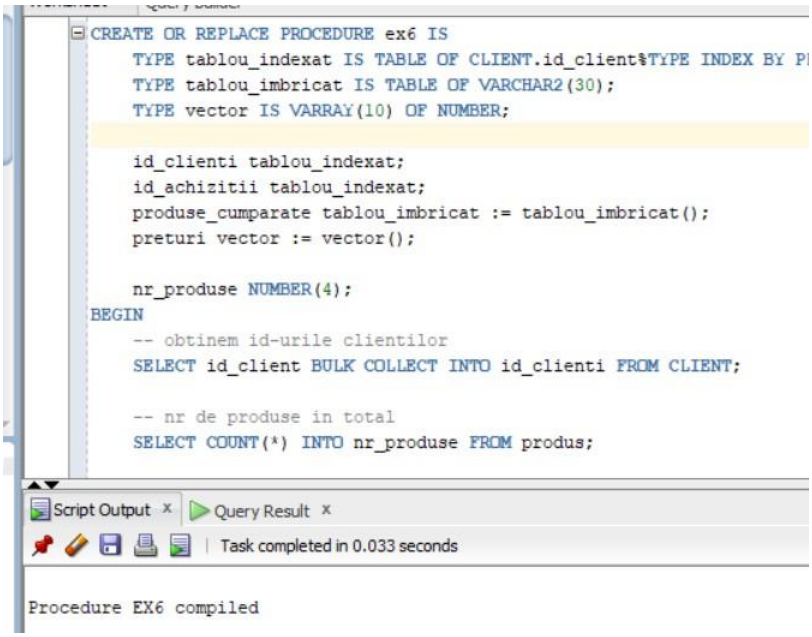


Figure 7: EX 6.

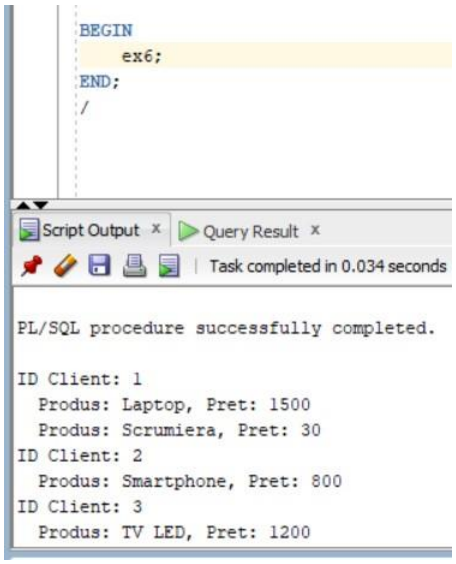

BEGIN
ex6;
END;
/
Script Output
Query Result
Task completed in 0.034 seconds
PL/SQL procedure successfully completed.
ID Client: 1
Produs: Laptop, Pret: 1500
Produs: Scrumiera, Pret: 30
ID Client: 2
Produs: Smartphone, Pret: 800
ID Client: 3
Produs: TV LED, Pret: 1200

Figure 8: EX 6.

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

```
/*
Sa se afiseze produsele tuturor magazinelor.
*/

CREATE OR REPLACE PROCEDURE ex7 AS


  CURSOR cursor_principal IS

    SELECT id_magazin, nume_magazin

    FROM MAGAZIN;


  v_parametru_cursor NUMBER;


  CURSOR cursor_intern(p_id_magazin NUMBER) IS

    SELECT id_achizitie, nume_produs, pret

    FROM PRODUS

    WHERE id_magazin = p_id_magazin;
```

```
BEGIN
   FOR principal_rec IN cursor_principal LOOP
      DBMS_OUTPUT.PUT_LINE('Magazin: ' || principal_rec.nume_magazin);
      v_parametru_cursor := principal_rec.id_magazin;
      FOR intern_rec IN cursor_intern(v_parametru_cursor) LOOP
         DBMS_OUTPUT.PUT_LINE('  Produs: ' || intern_rec.nume_produs || ', Pret: ' || intern_rec.pret);
      END LOOP;
   END LOOP;
END ex7;
/
BEGIN
   ex7;
END;
/
```

```
1  CREATE OR REPLACE PROCEDURE ex7 AS
2
3      CURSOR cursor_principal IS
4          SELECT id_magazin, nume_magazin
5          FROM MAGAZIN;
6
7      v_parametru_cursor NUMBER;
8
9      CURSOR cursor_intern(p_id_magazin NUMBER) IS
10         SELECT id_achizitie, nume_produs, pret
11         FROM PRODUS
12         WHERE id_magazin = p_id_magazin;
13
14 BEGIN
```

Script Output x | Query Result x

Task completed in 0.028 seconds

```
Procedure EX7 compiled
```

Figure 9: EX 7.

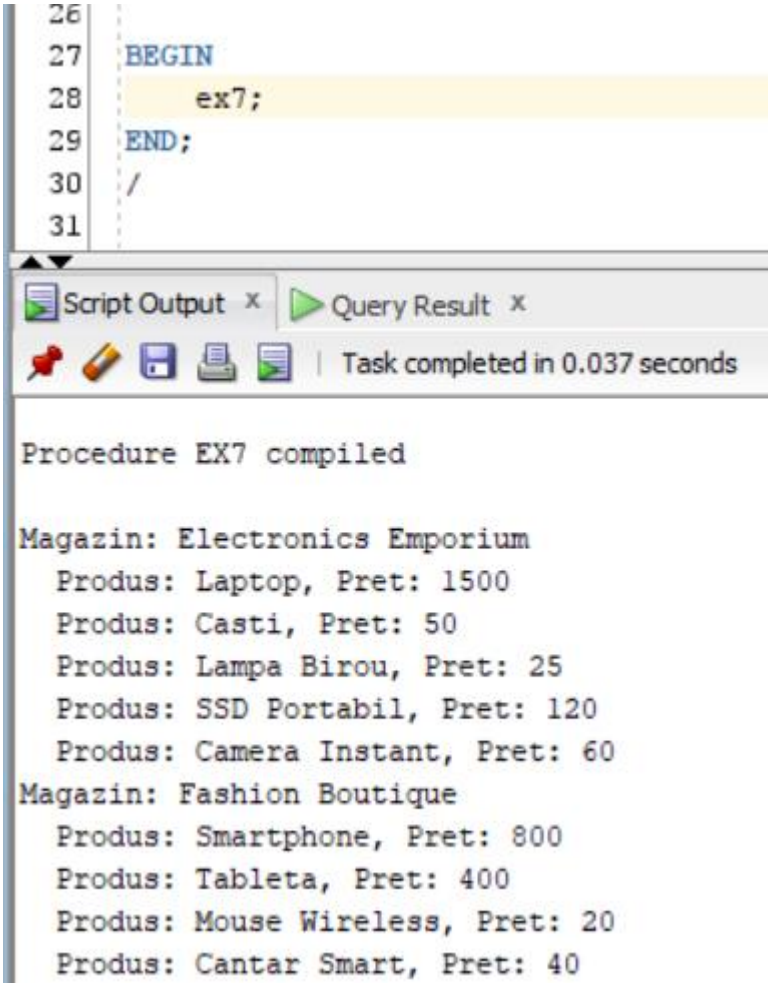


Figure 10: EX 7.

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

```
/*
Sa se afiseze numarul de angajati interni ale magazinelor care sunt angajati la n magazine, n fiind un
numar transmis ca parametru. In cazul in care numarul n transmis este <= 0 se va afisa mesajul “Numarul
de joburi trebuie sa fie mai mare decat 0”, iar in cazul in care nu exista niciun angajat care sa lucreze la fix
n joburi se va afisa mesajul “Nu exista angajati cu n joburi”.
*/

CREATE OR REPLACE FUNCTION ex8(nr_joburi IN NUMBER) RETURN NUMBER IS

   TYPE id_intern_array IS TABLE OF intern.id_intern%TYPE;

   v_angajati id_intern_array := id_intern_array();

   v_numar_angajati NUMBER := 0;



   PARAMETRU_INCORECT EXCEPTION;

   NU_EXISTA_ANGAJATI EXCEPTION;



BEGIN

   IF nr_joburi <= 0 THEN

      RAISE PARAMETRU_INCORECT;
```

```
  END IF;


  FOR angajat IN (
    SELECT i.id_intern, i.norma, m.id_magazin
    FROM intern i
    JOIN intern_magazin im ON i.id_intern = im.id_intern
    JOIN magazin m ON im.id_magazin = m.id_magazin
    WHERE i.id_intern IN (
      SELECT i2.id_intern
      FROM intern i2
      JOIN intern_magazin im2 ON i2.id_intern = im2.id_intern
      GROUP BY i2.id_intern
      HAVING COUNT(DISTINCT im2.id_magazin) = nr_joburi
    )
  ) LOOP
    DBMS_OUTPUT.PUT_LINE('Intern ID: ' || angajat.id_intern || ', Norma: ' || angajat.norma || ', Magazin
ID: ' || angajat.id_magazin);


    DECLARE
      v_exists BOOLEAN := FALSE;
    BEGIN
      FOR i IN 1..v_angajati.COUNT LOOP
```

```
          IF v_angajati(i) = angajat.id_intern THEN
            v_exists := TRUE;
            EXIT;
          END IF;
        END LOOP;

        IF NOT v_exists THEN
          v_angajati.EXTEND;
          v_angajati(v_angajati.LAST) := angajat.id_intern;
          v_numar_angajati := v_numar_angajati + 1;
        END IF;
      END;

  END LOOP;

  IF v_numar_angajati = 0 THEN
    RAISE NU_EXISTA_ANGAJATI;
  END IF;

  RETURN v_numar_angajati;

EXCEPTION
```

```
   WHEN PARAMETRU_INCORECT THEN

      DBMS_OUTPUT.PUT_LINE('Exceptie: Parametrul este incorect. Numarul de joburi trebuie sa fie mai
mare decat 0.');



   WHEN NU_EXISTA_ANGAJATI THEN

      DBMS_OUTPUT.PUT_LINE('Exceptie: Nu exista angajati cu ' || nr_joburi || ' joburi.');



END ex8;

/

DECLARE

   nr_joburi NUMBER := 2; -- TESTAM PT -1, 2, 5

   rezultat NUMBER;

BEGIN

   rezultat := ex8(nr_joburi);

   DBMS_OUTPUT.PUT_LINE('Nr angajati cu ' || nr_joburi || ' joburi : ' || rezultat);

END;

/
```

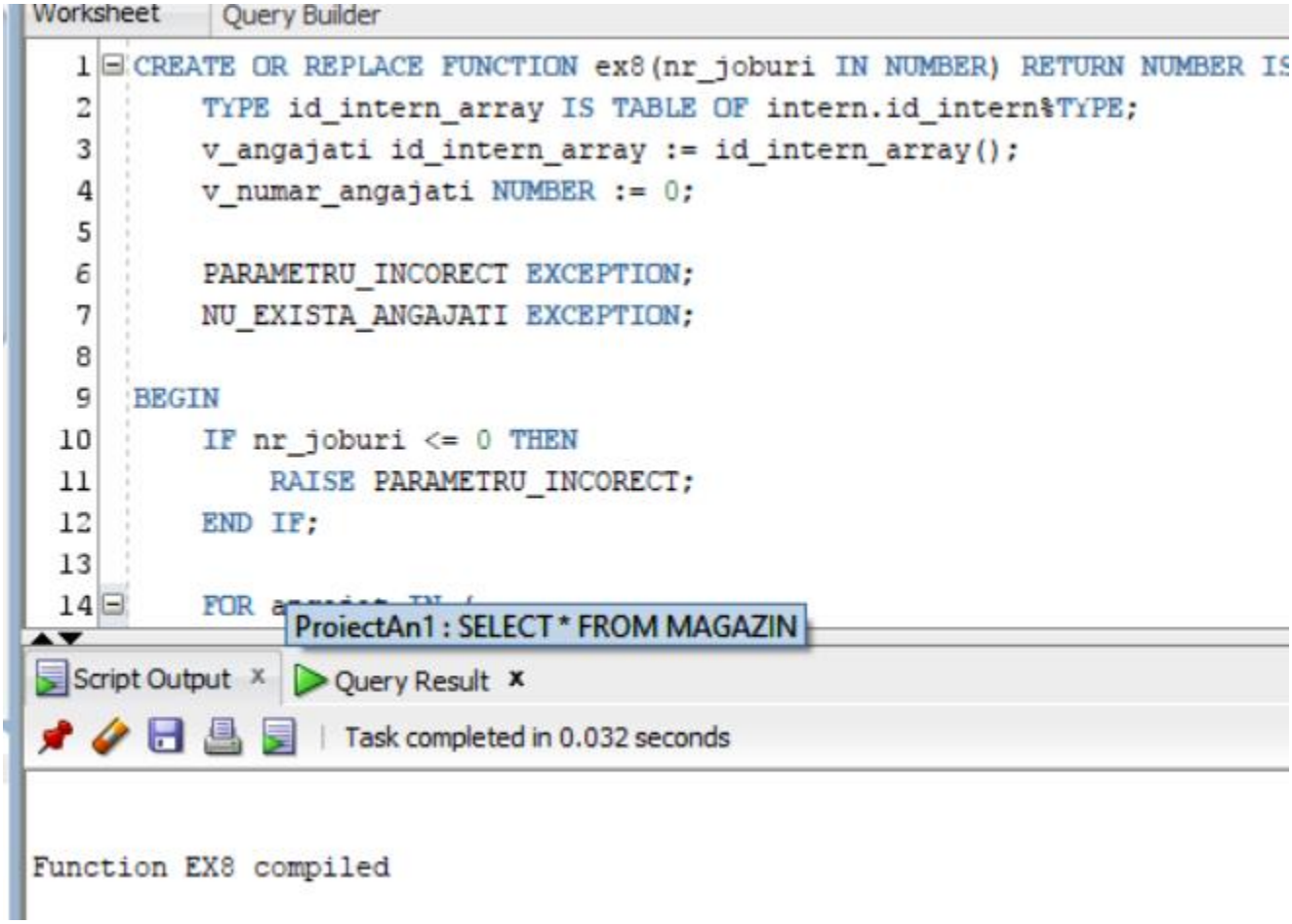


Figure 11: EX 8.

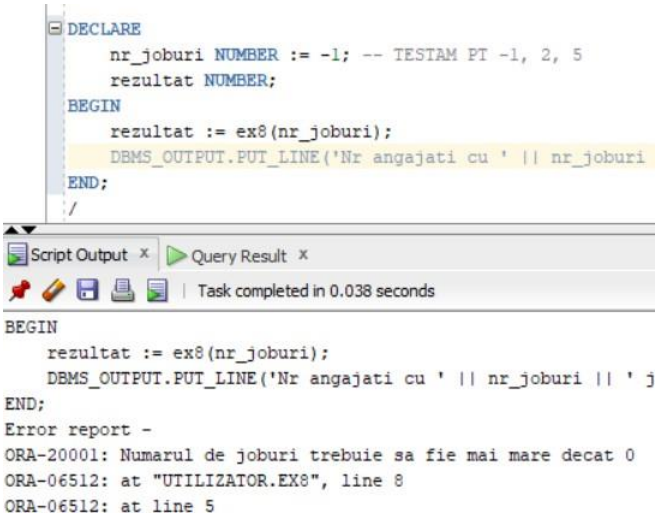


Figure 12: EX 8.

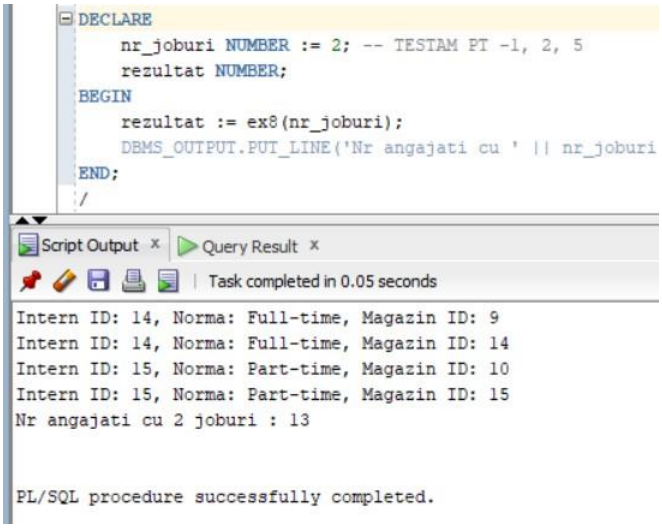


Figure 13: EX 8.

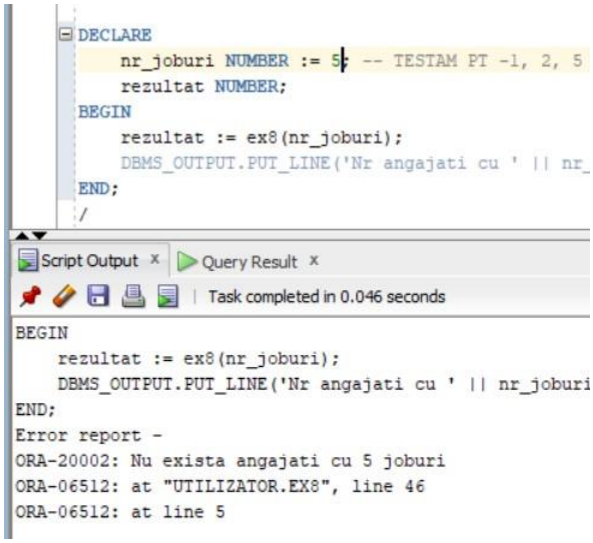


Figure 14: EX 8.

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

```
/*
Sa se afiseze magazinul din tara m (transmisa ca parametru) care are cele mai multe reclamatii. Daca
exista mai multe astfel de magazine se va declansa eroarea TOO_MANY_ROWS, iar daca nu exista niciun
magazine in tara respectiva se va declansa eroarea NOT_DATA_FOUND.
*/

CREATE OR REPLACE PROCEDURE ex9(p_tara VARCHAR2) AS

  CURSOR curRezultat IS

    SELECT per.nume, per.prenume, mag.nume_magazin, mal.tara, COUNT(rec.id_reclamatie) AS
total_reclamatii

    FROM PERSONAL per

    JOIN PROPRIETAR pro ON per.id_angajat = pro.id_angajat

    JOIN MAGAZIN mag ON pro.id_proprietar = mag.id_proprietar

    JOIN MALL mal ON mag.id_mall = mal.id_mall

    LEFT JOIN RECLAMATIE rec ON mag.id_magazin = rec.id_magazin

    WHERE mal.tara = p_tara

    GROUP BY per.nume, per.prenume, mag.nume_magazin, mal.tara
```

```
  ORDER BY COUNT(rec.id_reclamatie) DESC;


 vNumeProprietar VARCHAR2(30);

 vPrenumeProprietar VARCHAR2(30);

 vNumeMagazin VARCHAR2(30);

 vTaraMall VARCHAR2(50);

 vTotalReclamatii NUMBER;


 vNumeProprietar2 VARCHAR2(30);

 vPrenumeProprietar2 VARCHAR2(30);

 vNumeMagazin2 VARCHAR2(30);

 vTaraMall2 VARCHAR2(50);

 vTotalReclamatii2 NUMBER;


 vNumarRezultate NUMBER := 0;


BEGIN

 OPEN curRezultat;

 FETCH curRezultat INTO vNumeProprietar, vPrenumeProprietar, vNumeMagazin, vTaraMall,
vTotalReclamatii;

 IF curRezultat%FOUND THEN

  vNumarRezultate := 1;

  FETCH curRezultat INTO vNumeProprietar2, vPrenumeProprietar2, vNumeMagazin2, vTaraMall2,
```

```
vTotalReclamatii2;
   IF vTotalReclamatii = vTotalReclamatii2 THEN
     CLOSE curRezultat;
     RAISE TOO_MANY_ROWS;
   END IF;
  END IF;

  IF vNumarRezultate = 0 THEN
   CLOSE curRezultat;
   RAISE NO_DATA_FOUND;
  ELSE

   DBMS_OUTPUT.PUT_LINE('Primul magazin cu cele mai multe reclamatii:');
   DBMS_OUTPUT.PUT_LINE('Nume magazin: ' || vNumeMagazin);
   DBMS_OUTPUT.PUT_LINE('Proprietar: ' || vNumeProprietar || ' ' || vPrenumeProprietar);
   DBMS_OUTPUT.PUT_LINE('Tara mall-ului: ' || vTaraMall);
   DBMS_OUTPUT.PUT_LINE('Total reclamatii: ' || vTotalReclamatii);
  END IF;

EXCEPTION
  WHEN NO_DATA_FOUND THEN
   DBMS_OUTPUT.PUT_LINE('Nu exista mall-uri in aceasta tara');
```

```
 WHEN TOO_MANY_ROWS THEN
   DBMS_OUTPUT.PUT_LINE('Prea multe randuri returnate cu acelasi numar de reclamatii');
 WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('O eroare a apărut: ' || SQLERRM);
END ex9;
/

BEGIN
 ex9('Romania');
END;
/
BEGIN
 ex9('Spania');
END;
/
BEGIN
 ex9('Mongolia');
END;
/
```

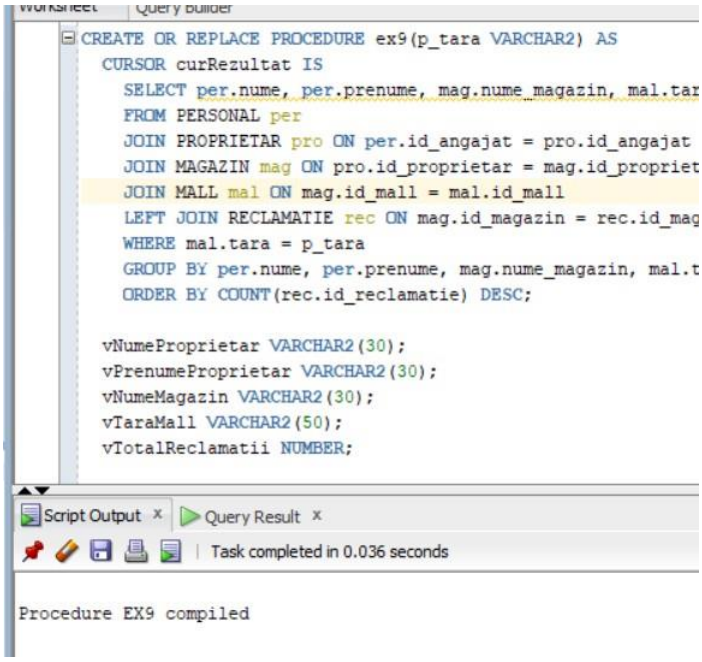


Figure 15: EX 9.

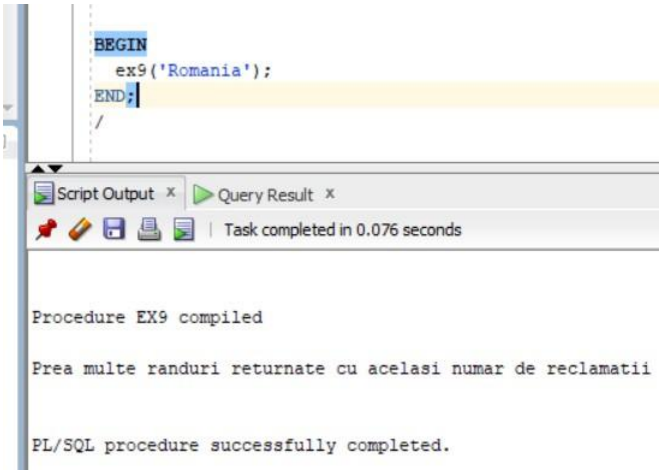


Figure 16: EX 9.

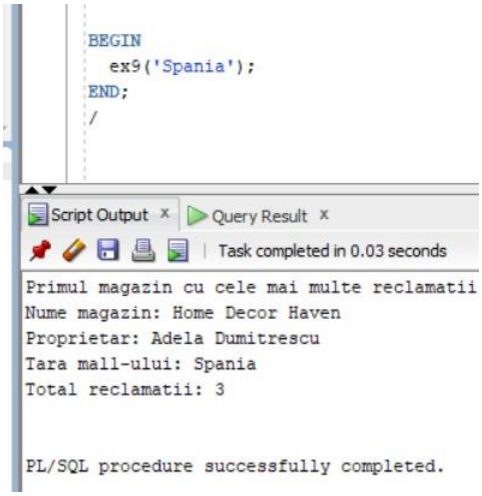


Figure 17: EX 9.

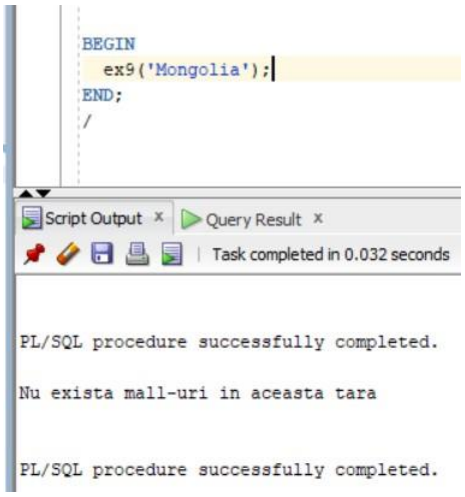


Figure 18: EX 9.

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

```
/*
Sa se creeze un trigger de tip LMD care sa nu permita inserarea de produse intre orele 22:00 si 6:00. (In screenshot se poate observa inclusive ora la care a fost incercata inserarea).
*/

CREATE OR REPLACE TRIGGER ex10

BEFORE INSERT ON PRODUS

BEGIN

  IF TO_CHAR(SYSDATE, 'HH24') BETWEEN 10 AND 23 OR TO_CHAR(SYSDATE, 'HH24') BETWEEN 0 AND 6 THEN

    RAISE_APPLICATION_ERROR(-20001, 'Nu se pot insera produse între ora 22:00 și 06:00.');

  END IF;

END;

/

COMMIT;

INSERT INTO PRODUS VALUES(53, 200, 'Masinuta', 1, 10);
```

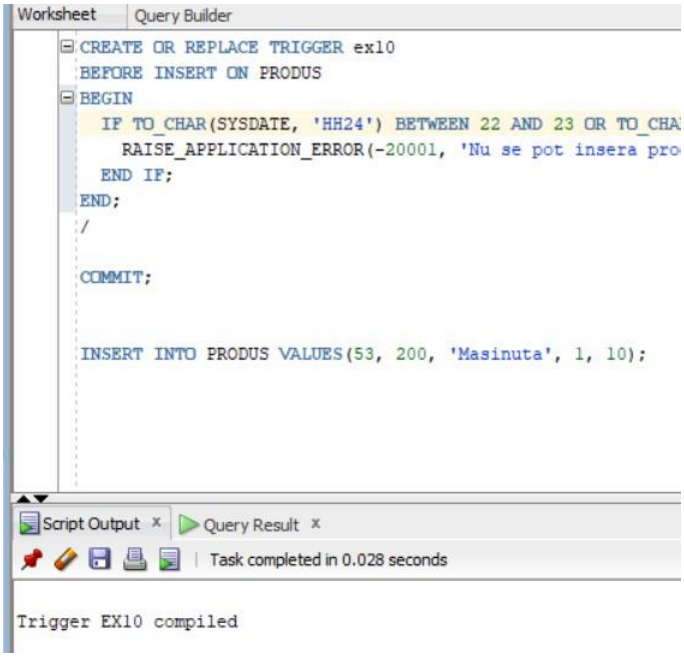


Figure 19: EX 10.

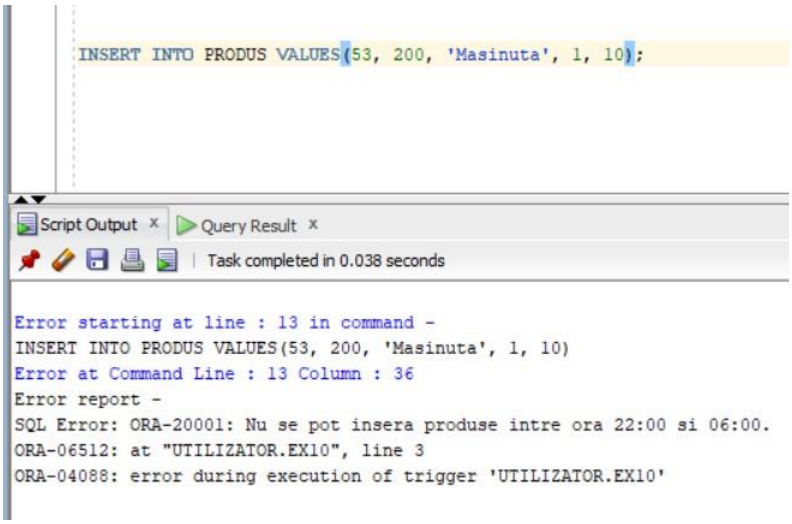


Figure 20: EX 10.

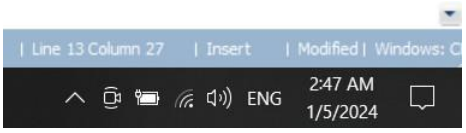


Figure 21: EX 10.

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

```
/*
Sa se creeze un trigger de tip LMD care sa nu permita marirea salariului oricarui angajat astfel incat sa depaseasca dublul salariului deja existent.
*/

CREATE OR REPLACE TRIGGER ex11

BEFORE UPDATE OF salariu ON personal

FOR EACH ROW

BEGIN

   IF (:NEW.salariu >= :OLD.salariu * 2) THEN

      RAISE_APPLICATION_ERROR(-20002,'Salariu este prea mare!');

   END IF;

END;

/

UPDATE personal

SET salariu = 9999

WHERE id_angajat = 1;
```

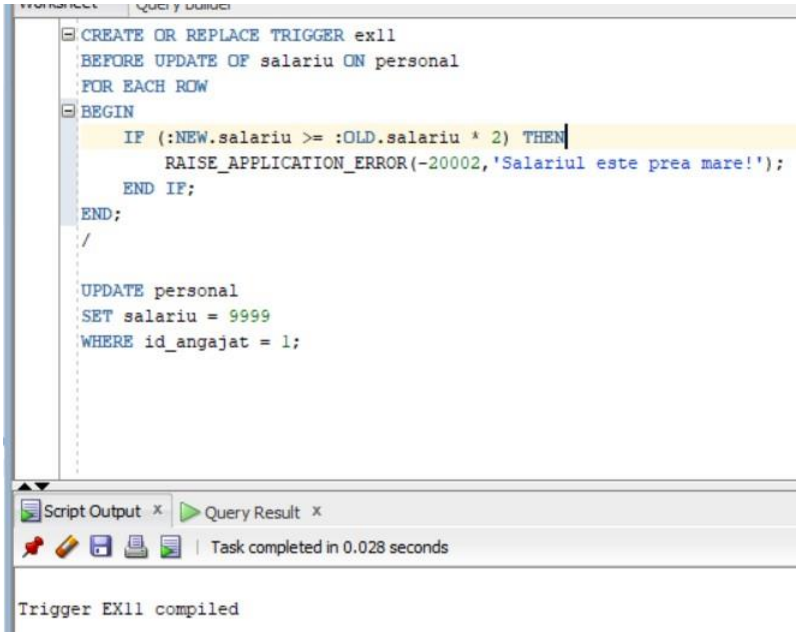


Figure 22: EX 11.

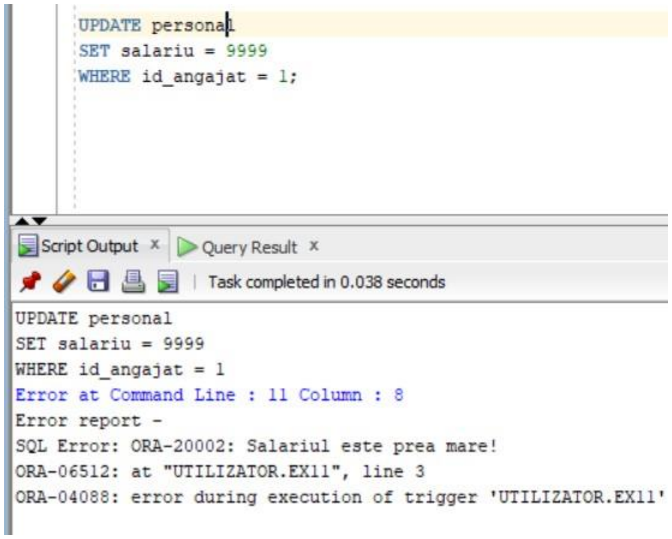


Figure 23: EX 11.

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

```
/*
Sa se creeze un trigger de tip LDD, care sa contorizeze in tabelul “istoric” orice creare, stergere sau
modificarea a unui tabel. In acest tabel se va contoriza utilizatorul care a incercat sa creeze / stearga /
modice, numele bazei de date, operatia facuta, numele obiectului si data.
*/

CREATE TABLE istoric

  (utilizator VARCHAR2(30),

   nume_bd VARCHAR2(50),

   eveniment VARCHAR2(20),

   nume_obiect VARCHAR2(30),

   data DATE);



CREATE OR REPLACE TRIGGER ex12

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

   INSERT INTO istoric

   VALUES (SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT,

   SYS.DICTIONARY_OBJ_NAME, SYSDATE);

END;

/
```

```
CREATE TABLE exemplu1 (
   id_ex NUMBER(4) PRIMARY KEY);

DROP TABLE exemplu1;

select * from istoric;
```

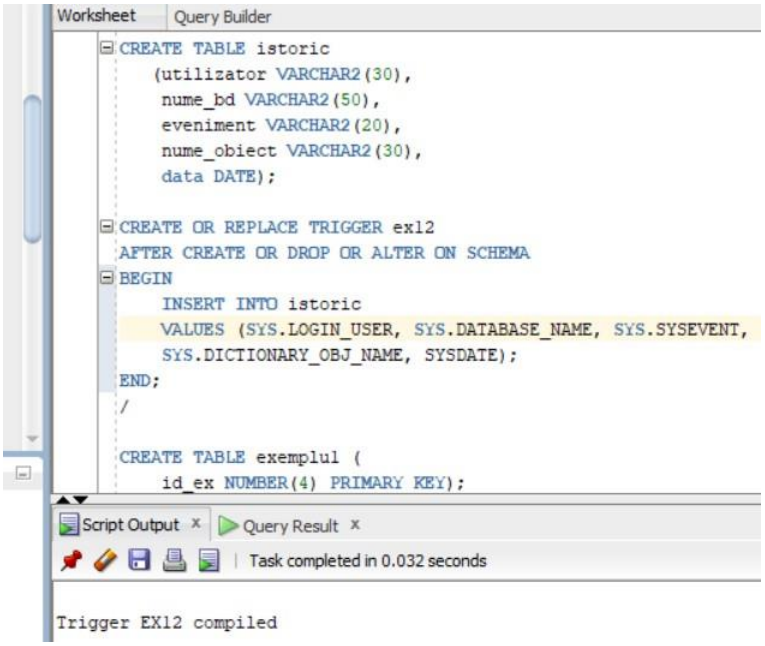


Figure 24: EX 12.

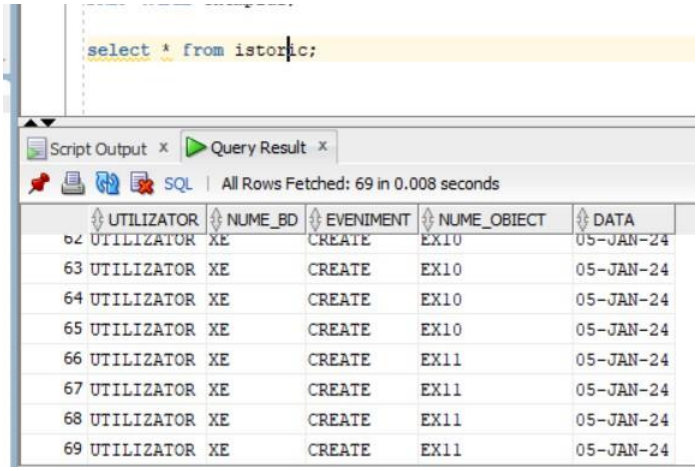


Figure 25: EX 12.