

PROIECT BAZE DE DATE

Sistem de Management Bancar

Serban Silviu

Grupa: 1065

Seria: E

Cuprins

Descrierea bazei de date	3
Cod creare tabele:.....	7
Model Relațional pentru Sistemul de Management Bancar.....	10
Populare Tabele:.....	11
Alte operații LDD:	16
Alte operații LMD:	19
Exemple de interogări	25

Descrierea bazei de date

Baza de date reprezintă o componentă fundamentală a unui sistem de management bancar, concepută pentru a stoca și organiza datele necesare desfășurării activităților zilnice ale băncii. Acest sistem facilitează gestionarea operațiunilor financiare și menținerea relațiilor cu clienții, oferind un flux de informații precis și eficient între clienți, conturi, produse financiare și sucursale.

Baza de date este structurată în mai multe tabele interconectate, fiecare având un rol bine definit. Tabela dedicată tranzacțiilor stochează detalii despre retrageri, depuneri și transferuri între conturi, facilitând urmărirea fluxurilor financiare. Conturile bancare sunt gestionate prin tabele care includ informații despre solduri, tipurile de conturi și legătura acestora cu clienții. Depozitele bancare și creditele sunt administrate prin tabele specifice, care permit monitorizarea sumelor economisite sau împrumutate.

De asemenea, sistemul înregistrează informații despre cardurile asociate conturilor, precum numărul cardului, limitele de credit și datele de valabilitate, asigurând o administrare completă a produselor financiare. Toate aceste elemente contribuie la crearea unei baze de date centralizate, care oferă suport pentru raportare, audit și respectarea cerințelor legale, făcând activitățile bancare mai eficiente și mai transparente.

Tabele bazei de date:

Tabela SUCURSALE:

- **Cheie primară:** ID_Sucursală
- **Atribute:**
 - *Nume_Sucursală* (VARCHAR) – Numele sucursalei.
 - *Adresa* (VARCHAR) – Adresa completă a sucursalei.
 - *Telefon* (VARCHAR) – Numărul de telefon al sucursalei.
- **Descriere:** Reține informații despre locațiile fizice ale băncii unde sunt gestionate operațiunile.

Tabela CLIENTI:

- **Cheie primară:** ID_Client
- **Atribute:**
 - *Nume* (VARCHAR) – Numele clientului.
 - *Prenume* (VARCHAR) – Prenumele clientului.
 - *CNP* (VARCHAR) – Codul Numeric Personal al clientului.
 - *Adresa* (VARCHAR) – Adresa clientului.
 - *Telefon* (VARCHAR) – Numărul de telefon al clientului.
 - *Email* (VARCHAR) – Adresa de email a clientului.
 - *Data_Înregistrării* (DATE) – Data înregistrării clientului în sistem.
- **Descriere:** Reține informațiile personale ale clienților pentru identificare și contact.

Tabela CONTURI:

- **Cheie primară:** ID_Cont

- **Atribute:**
 - *ID_Client* (NUMBER, **FK**, referă la **Clienți**) – ID-ul clientului care deține contul.
 - *ID_Tip_Cont* (NUMBER, **FK**, referă la **Tipuri_Conturi**) – ID-ul tipului de cont (Curent, Economii, Credit etc.).
 - *Sold* (NUMBER) – Soldul curent al contului.
 - *Data_Deschiderii* (DATE) – Data deschiderii contului.
 - *Stare* (VARCHAR) – Starea contului (Activ, Suspuspendat, Închis).
 - *ID_Sucursala* (NUMBER, **FK**, referă la **Sucursale**) – ID-ul sucursalei unde a fost deschis contul.
- **Descriere:** Reține informații despre conturile bancare ale clienților, inclusiv detalii financiare, tipul de cont și locația deschiderii.

Tabela TIPURI_CONTURI

- **Cheie primară:** *ID_Tip_Cont*
- **Atribute:**
 - *Denumire* (VARCHAR) – Numele tipului de cont (de exemplu: Curent, Economii, Credit).
 - *Dobândă* (NUMBER) – Rata dobânzii asociată acestui tip de cont.
 - *Limita_Descoperire* (NUMBER) – Suma maximă permisă pentru overdraft (dacă este aplicabil).
- **Descriere:** Definește și stochează caracteristicile fiecărui tip de cont oferit de bancă. Aceasta permite o configurare ușoară și flexibilă a ofertelor băncii pentru diferite tipuri de conturi.

Tabela ANGAJĂȚI:

- **Cheie primară:** *ID_Angajat*
- **Atribute:**
 - *Prenume* (VARCHAR) – Prenumele angajatului.
 - *Nume* (VARCHAR) – Numele angajatului.
 - *CNP* (VARCHAR) – Cod Numeric Personal al angajatului.
 - *ID_Sucursală* (NUMBER, FK, referă la **Sucursale**) – ID-ul sucursalei unde lucrează angajatul.
 - *Pozиie* (VARCHAR) – Funcția angajatului (Manager, Casier, etc.).
 - *Data_Angajării* (DATE) – Data în care a fost angajat.
 - *Salariu* (NUMBER) – Valoarea salariului angajatului.
 - *Telefon* (VARCHAR) – Numărul de telefon.
 - *Email* (VARCHAR) – Adresa de email a angajatului.
- **Descriere:** Reține informațiile personale și detaliile contractuale ale angajaților băncii.

Tabela TRANZACȚII:

- **Cheie primară:** *ID_Tranzactie*
- **Atribute:**
 - *ID_Cont_Sursă* (NUMBER, **FK**, referă la **Conturi**) – ID-ul contului sursă.
 - *ID_Cont_Destinație* (NUMBER, **FK**, referă la **Conturi**) – ID-ul contului destinație.
 - *Sumă* (NUMBER) – Valoarea tranzacției.
 - *Data_Tranzacției* (TIMESTAMP) – Data și ora tranzacției.
 - *Tip_Tranzacție* (VARCHAR) – Tipul tranzacției (Transfer, Depunere, Retragere).
- **Descriere:** Reține detalii despre tranzacțiile financiare dintre conturile băncii.

Tabela CREDITE:

- **Cheie primară:** ID_Credit
- **Atribute:**
 - *ID_Client* (NUMBER, **FK**, referă la **Clienți**) – ID-ul clientului care a luat creditul.
 - *Suma_Aprobata* (NUMBER) – Suma totală aprobată pentru credit.
 - *Rata_Dobânzii* (NUMBER) – Rata dobânzii asociată creditului.
 - *Data_Acordării* (DATE) – Data aprobării creditului.
 - *Data_Scădenței* (DATE) – Data scadenței pentru rambursarea completă.
 - *Status_Credit* (VARCHAR) – Statusul creditului (În derulare, Achitat, Restant).
- **Descriere:** Reține informații despre creditele acordate clienților.

Tabela DEPOZITE:

- **Cheie primară:** ID_Depozit
- **Atribute:**
 - *ID_Cont* (NUMBER, **FK**, referă la **Conturi**) – ID-ul contului asociat depozitului.
 - *Suma_Depozit* (NUMBER) – Suma depusă.
 - *Rata_Dobânzii* (NUMBER) – Rata dobânzii asociată depozitului.
 - *Data_Depunere* (DATE) – Data depunerii.
 - *Data_Scadenta* (DATE) – Data scadenței.
 - *Status_Depozit* (VARCHAR) – Statusul depozitului (În derulare, Închis).
- **Descriere:** Reține detalii despre depozitele constituite de clienți în conturile lor.

Tabela CARDURI

- **Cheie primară:** ID_Card
- **Atribute:**
 - *Numar_Card* (VARCHAR) – Numărul unic al cardului (16 cifre).
 - *Tip_Card* (VARCHAR) – Tipul cardului (de exemplu: Debit, Credit).
 - *Data_Emitere* (DATE) – Data emiterii cardului.
 - *Data_Expirare* (DATE) – Data expirării cardului.
 - *Limita_Credit* (NUMBER) – Limita de credit disponibilă (doar pentru carduri de credit).
 - *ID_Cont* (NUMBER, **FK**, referă la **Conturi**) – ID-ul contului bancar asociat cardului.
- **Descriere:** Stocăreză informațiile despre cardurile emise clienților băncii. Fiecare card este asociat unui cont bancar și poate fi de tip debit sau credit.

Relațiile între tabele:

- **Clienți-Conturi:** Un client poate avea mai multe conturi bancare. id_client din Conturi face referință la id_client din Clienți. Relație 1:N.
- **Conturi-Tranzacții:** Un cont poate fi implicat în mai multe tranzacții. id_cont_sursă și id_cont_destinație din Tranzacții fac referință la id_cont din Conturi. Relație 1:N.
- **Conturi-Tipuri_Conturi:** Fiecare cont este asociat unui tip de cont. id_tip_cont din Conturi face referință la id_tip_cont din Tipuri_Conturi. Relație N:1.
- **Sucursale-Conturi:** O sucursală poate avea mai multe conturi deschise. id_sucursală din Conturi face referință la id_sucursală din Sucursale. Relație 1:N.
- **Sucursale-Angajați:** O sucursală poate avea mulți angajați. id_sucursală din Angajati face referință la id_sucursală din Sucursale. Relație 1:N.
- **Conturi-Carduri:** Un cont poate avea mai multe carduri asociate. id_cont din Carduri face referință la id_cont din Conturi. Relație 1:N.
- **Clienți-Credite:** Un client poate avea mai multe credite. id_client din Credite face referință la id_client din Clienți. Relație 1:N.
- **Conturi-Depozite:** Un cont poate avea mai multe depozite. id_cont din Depozite face referință la id_cont din Conturi. Relație 1:N.

Cod creare tabele:

```
CREATE TABLE Sucursala (
    ID_Sucursala NUMBER(10) PRIMARY KEY,
    Nume_Sucursala VARCHAR2(50) NOT NULL,
    Adresa VARCHAR2(100),
    Telefon VARCHAR2(15)
);
```

```
CREATE TABLE Clienti (
    ID_Client NUMBER(10) PRIMARY KEY,
    Nume VARCHAR2(50) NOT NULL,
    Prenume VARCHAR2(50) NOT NULL,
    CNP VARCHAR2(13) UNIQUE NOT NULL,
    Adresa VARCHAR2(100),
    Telefon VARCHAR2(15),
    Email VARCHAR2(50),
    Data_Inregistrarii DATE DEFAULT SYSDATE
);
```

```
CREATE TABLE Tipuri_Conturi (
    ID_Tip_Cont NUMBER(10) PRIMARY KEY,
    Denumire VARCHAR2(20) NOT NULL,
    Dobanda NUMBER(5, 2),
    Limita_Descoperire NUMBER(12, 2)
);
```

```
CREATE TABLE Conturi (
    ID_Cont NUMBER(10) PRIMARY KEY,
    ID_Client NUMBER(10) NOT NULL,
    ID_Tip_Cont NUMBER(10) NOT NULL,
    Sold NUMBER(12, 2) DEFAULT 0,
    Data_Deschiderii DATE DEFAULT SYSDATE,
    Stare VARCHAR2(20),
    ID_Sucursala NUMBER(10) NOT NULL,
    CONSTRAINT FK_Conturi_Clienti FOREIGN KEY (ID_Client) REFERENCES Clienti(ID_Client),
    CONSTRAINT FK_Conturi_Tipuri FOREIGN KEY (ID_Tip_Cont) REFERENCES
    Tipuri_Conturi(ID_Tip_Cont),
    CONSTRAINT FK_Conturi_Sucursala FOREIGN KEY (ID_Sucursala) REFERENCES
    Sucursala(ID_Sucursala)
);
```

```

CREATE TABLE Carduri (
    ID_Card NUMBER(10) PRIMARY KEY,
    Numar_Card VARCHAR2(16) UNIQUE NOT NULL,
    Tip_Card VARCHAR2(20) NOT NULL,
    Data_Emitere DATE DEFAULT SYSDATE,
    Data_Expirare DATE NOT NULL,
    Limita_Credit NUMBER(12, 2),
    ID_Cont NUMBER(10) NOT NULL,
    CONSTRAINT FK_Carduri_Conturi FOREIGN KEY (ID_Cont) REFERENCES Conturi(ID_Cont)
);

```

```

CREATE TABLE Tranzactii (
    ID_Tranzactie NUMBER(12) PRIMARY KEY,
    ID_Cont_Sursa NUMBER(10),
    ID_Cont_Destinatie NUMBER(10),
    Suma NUMBER(12, 2) NOT NULL,
    Data_Tranzactiei TIMESTAMP DEFAULT SYSTIMESTAMP,
    Tip_Tranzactie VARCHAR2(20),
    CONSTRAINT FK_Tranzactii_Cont_Sursa FOREIGN KEY (ID_Cont_Sursa) REFERENCES
    Conturi(ID_Cont),
    CONSTRAINT FK_Tranzactii_Cont_Dest FOREIGN KEY (ID_Cont_Destinatie) REFERENCES
    Conturi(ID_Cont)
);

```

```

CREATE TABLE Angajati (
    ID_Angajat NUMBER(10) PRIMARY KEY,
    Prenume VARCHAR2(50) NOT NULL,
    Nume VARCHAR2(50) NOT NULL,
    CNP VARCHAR2(13) UNIQUE NOT NULL,
    ID_Sucursala NUMBER(10) NOT NULL,
    Pozitie VARCHAR2(30) NOT NULL,
    Data_Angajarii DATE DEFAULT SYSDATE,
    Salariu NUMBER(10, 2) NOT NULL,
    Telefon VARCHAR2(15),
    Email VARCHAR2(50) UNIQUE,
    CONSTRAINT FK_Angajati_Sucursala FOREIGN KEY (ID_Sucursala) REFERENCES
    Sucursala(ID_Sucursala)
);

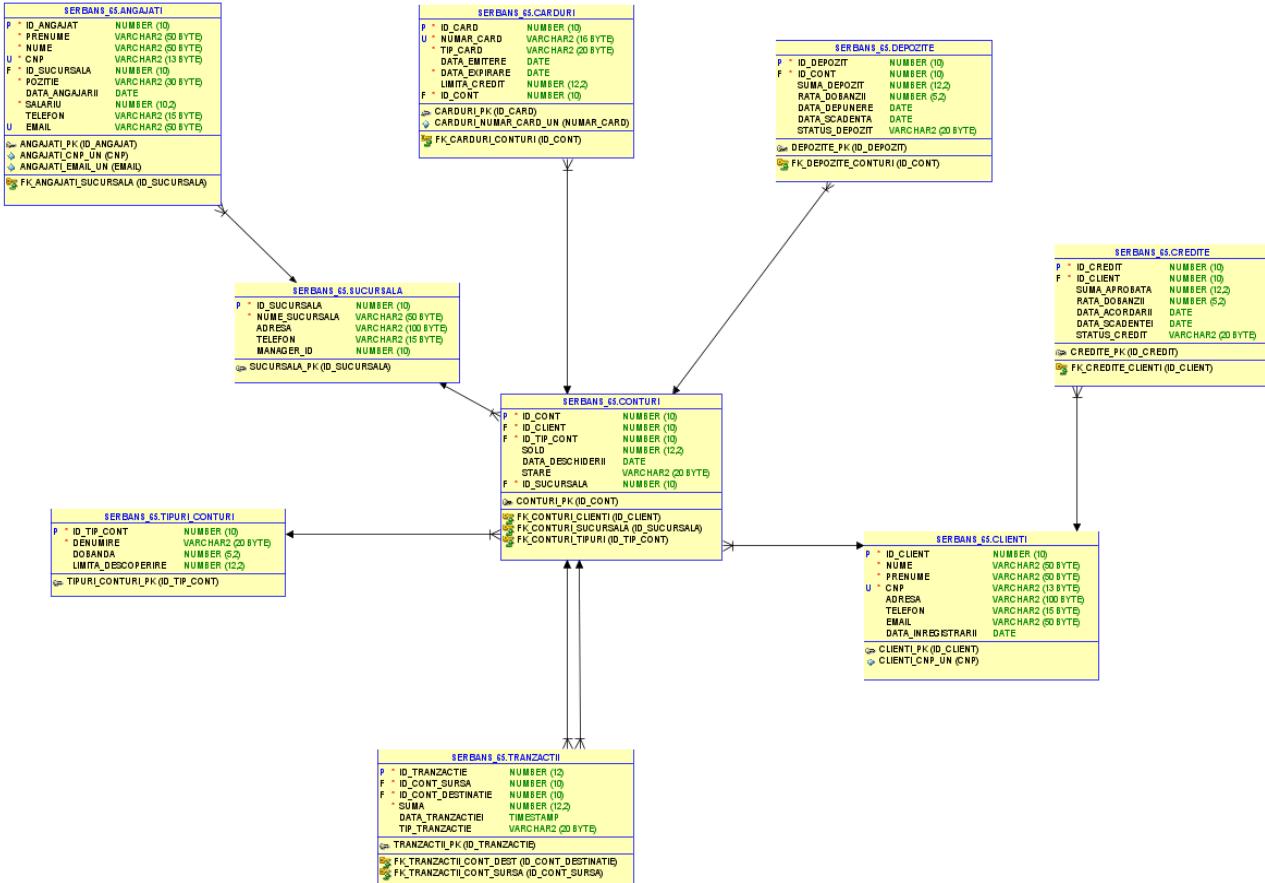
```

```
CREATE TABLE Credite (
    ID_Credit NUMBER(10) PRIMARY KEY,
    ID_Client NUMBER(10) NOT NULL,
    Suma_Aprobata NUMBER(12, 2),
    Rata_Dobanzii NUMBER(5, 2),
    Data_Acordarii DATE DEFAULT SYSDATE,
    Data_Scadentei DATE,
    Status_Credit VARCHAR2(20),
    CONSTRAINT FK_Credite_Clienti FOREIGN KEY (ID_Client) REFERENCES Clienti(ID_Client)
);
```

```
CREATE TABLE Depozite (
    ID_Depozit NUMBER(10) PRIMARY KEY,
    ID_Cont NUMBER(10) NOT NULL,
    Suma_Depozit NUMBER(12, 2),
    Rata_Dobanzii NUMBER(5, 2),
    Data_Depunere DATE DEFAULT SYSDATE,
    Data_Scadenta DATE,
    Status_Depozit VARCHAR2(20),
    CONSTRAINT FK_Depozite_Conturi FOREIGN KEY (ID_Cont) REFERENCES Conturi(ID_Cont)
);
```

Model Relațional pentru Sistemul de Management Bancar

Generat după crearea tabelelor și definirea relațiilor dintre entități.



Populare Tabele:

```
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (1, 'Bucuresti', 'Strada Principala 1', '0211234567');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (2, 'Cluj', 'Strada Nordului 12', '0212233445');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (3, 'Timisoara', 'Strada Sudului 45', '0214455667');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (4, 'Iasi', 'Strada Estului 3', '0215566778');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (5, 'Constanta', 'Strada Vestului 8', '0216677889');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (6, 'Brasov', 'Bulevardul Bucuresti 100', '0217788990');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (7, 'Sibiu', 'Strada Clujului 22', '0218899001');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (8, 'Craiova', 'Strada Iasului 15', '0219001123');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (9, 'Arad', 'Strada Timisoarei 30', '0211234568');
INSERT INTO Sucursala (ID_Sucursala, Nume_Sucursala, Adresa, Telefon)
VALUES (10, 'Oradea', 'Strada Marii 50', '0212345678');

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (0, 'Silviu', 'Serban', '1900101123456', 1, 'Director', SYSDATE, 10000.00, '0745123456', 'silviu.serban@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (1, 'Ion', 'Popescu', '5900101123456', 1, 'Manager', SYSDATE, 5000.00, '0740123456', 'ion.popescu@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (2, 'Maria', 'Ionescu', '2910101123456', 1, 'Caser', SYSDATE, 3000.00, '0741123456', 'maria.ionescu@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (3, 'Andrei', 'Georgescu', '1920101123456', 1, 'Consultant Financiar', SYSDATE, 3500.00, '0742123456',
'andrei.georgescu@gmail.com');

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (4, 'Ana', 'Dumitrescu', '2930101123456', 2, 'Manager', SYSDATE, 5000.00, '0743123456', 'ana.dumitrescu@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (5, 'Mihai', 'Vasilescu', '1940101123456', 2, 'Caser', SYSDATE, 3000.00, '0744123456', 'mihai.vasilescu@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (6, 'Elena', 'Popa', '2950101123456', 2, 'Consultant Financiar', SYSDATE, 3500.00, '0745123456', 'elena.popa@yahoo.com');

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (7, 'Cristian', 'Stan', '1960101123456', 3, 'Manager', SYSDATE, 5000.00, '0746123456', 'cristian.stan@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (8, 'Alina', 'Marinescu', '2970101123456', 3, 'Caser', SYSDATE, 3000.00, '0747123456', 'alina.marinescu@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (9, 'George', 'Iliescu', '1980101123456', 3, 'Consultant Financiar', SYSDATE, 3500.00, '0748123456', 'george.iliescu@gmail.com');

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (10, 'Raluca', 'Stanciu', '2990101123456', 4, 'Manager', SYSDATE, 5000.00, '0749123456', 'raluca.stanciu@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (11, 'Alex', 'Moldoveanu', '2000101123456', 4, 'Caser', SYSDATE, 3000.00, '0750123456', 'alex.moldoveanu@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (12, 'Diana', 'Tudor', '2010101123456', 4, 'Consultant Financiar', SYSDATE, 3500.00, '0751123456', 'diana.tudor@yahoo.com');

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (13, 'Sorin', 'Radulescu', '2020101123456', 5, 'Manager', SYSDATE, 5000.00, '0752123456', 'sorin.radulescu@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (14, 'Simona', 'Calinescu', '2030101123456', 5, 'Caser', SYSDATE, 3000.00, '0753123456', 'simona.calinescu@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (15, 'Victor', 'Grigorescu', '2040101123456', 5, 'Consultant Financiar', SYSDATE, 3500.00, '0754123456',
'vector.grigorescu@gmail.com');

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (16, 'Adrian', 'Popa', '2050101123456', 6, 'Manager', SYSDATE, 5000.00, '0755123456', 'adrian.popa@gmail.com');
```

```

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (17, 'Gabriel', 'Marin', '2060101123456', 6, 'Casier', SYSDATE, 3000.00, '0756123456', 'gabriel.marin@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (18, 'Roxana', 'Enache', '2070101123456', 6, 'Consultant Financiar', SYSDATE, 3500.00, '0757123456', 'roxana.enache@gmail.com');


```

```

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (19, 'Oana', 'Mihai', '2080101123456', 7, 'Manager', SYSDATE, 5000.00, '0758123456', 'oana.mihai@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (20, 'Vlad', 'Tudor', '2090101123456', 7, 'Casier', SYSDATE, 3000.00, '0759123456', 'vlad.tudor@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (21, 'Iulia', 'Dobre', '2100101123456', 7, 'Consultant Financiar', SYSDATE, 3500.00, '0760123456', 'iulia.dobre@yahoo.com');


```

```

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (22, 'Marius', 'Popescu', '2110101123456', 8, 'Manager', SYSDATE, 5000.00, '0761123456', 'marius.popescu@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (23, 'Carmen', 'Toma', '2120101123456', 8, 'Casier', SYSDATE, 3000.00, '0762123456', 'carmen.toma@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (24, 'Alin', 'Georgescu', '2130101123456', 8, 'Consultant Financiar', SYSDATE, 3500.00, '0763123456', 'alin.georgescu@gmail.com');


```

```

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (25, 'Dan', 'Iordache', '2140101123456', 9, 'Manager', SYSDATE, 5000.00, '0764123456', 'dan.iordache@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (26, 'Roxana', 'Stoian', '2150101123456', 9, 'Casier', SYSDATE, 3000.00, '0765123456', 'roxana.stoian@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (27, 'Cristina', 'Dumitru', '2160101123456', 9, 'Consultant Financiar', SYSDATE, 3500.00, '0766123456',
'cristina.dumitru@gmail.com');


```

```

INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (28, 'Alexandru', 'Serban', '2170101123456', 10, 'Manager', SYSDATE, 5000.00, '0767123456', 'alexandru.serban@gmail.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (29, 'Bianca', 'Enescu', '2180101123456', 10, 'Casier', SYSDATE, 3000.00, '0768123456', 'bianca.enescu@yahoo.com');
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
VALUES (30, 'Razvan', 'Dinu', '2190101123456', 10, 'Consultant Financiar', SYSDATE, 3500.00, '0769123456', 'razvan.dinu@gmail.com');


```

```

INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (1, 'Popescu', 'Ion', '1900101123456', 'Strada Principala 1', '0740123456', 'ion.popescu@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (2, 'Ionescu', 'Maria', '2910101123456', 'Strada Lalelelor 12', '0741123456', 'maria.ionescu@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (3, 'Georgescu', 'Andrei', '1920101123456', 'Strada Florilor 8', '0742123456', 'andrei.georgescu@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (4, 'Dumitrescu', 'Ana', '2930101123456', 'Strada Castanilor 15', '0743123456', 'ana.dumitrescu@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (5, 'Vasilescu', 'Mihai', '1940101123456', 'Strada Libertatii 20', '0744123456', 'mihai.vasilescu@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (6, 'Popa', 'Elena', '2950101123456', 'Strada Independentei 25', '0745123456', 'elena.popa@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (7, 'Stan', 'Cristian', '1960101123456', 'Strada Viitorului 30', '0746123456', 'cristian.stan@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (8, 'Marinescu', 'Alina', '2970101123456', 'Strada Sperantei 35', '0747123456', 'alina.marinescu@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (9, 'Iliescu', 'George', '1980101123456', 'Strada Fericirii 40', '0748123456', 'george.iliescu@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (10, 'Stanciu', 'Raluca', '2990101123456', 'Strada Visului 45', '0749123456', 'raluca.stanciu@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (11, 'Moldoveanu', 'Alex', '2000101123456', 'Strada Bucuriei 50', '0750123456', 'alex.moldoveanu@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (12, 'Tudor', 'Diana', '2010101123456', 'Strada Copilului 55', '0751123456', 'diana.tudor@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (13, 'Radulescu', 'Sorin', '2020101123456', 'Strada Curcubeului 60', '0752123456', 'sorin.radulescu@gmail.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (14, 'Calinescu', 'Simona', '2030101123456', 'Strada Prieteniei 65', '0753123456', 'simona.calinescu@yahoo.com', SYSDATE);
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (15, 'Grigorescu', 'Victor', '2040101123456', 'Strada Pacii 70', '0754123456', 'victor.grigorescu@gmail.com', SYSDATE);


```

```

INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (1, 1, 5000.00, 5.00, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (2, 2, 10000.00, 4.50, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (3, 3, 7500.00, 6.00, SYSDATE, ADD_MONTHS(SYSDATE, 18), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (4, 4, 15000.00, 3.50, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (5, 5, 2000.00, 5.50, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (6, 6, 8000.00, 4.00, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (7, 7, 12000.00, 3.75, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (8, 8, 9000.00, 5.25, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (9, 9, 10000.00, 4.00, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (10, 10, 5000.00, 5.00, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (11, 11, 7000.00, 4.50, SYSDATE, ADD_MONTHS(SYSDATE, 18), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (12, 12, 11000.00, 3.75, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (13, 13, 15000.00, 3.50, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (14, 14, 20000.00, 4.75, SYSDATE, ADD_MONTHS(SYSDATE, 48), 'Activ');
INSERT INTO Credite (ID_Credit, ID_Client, Suma_Aprobata, Rata_Dobanzii, Data_Acordarii, Data_Scadentei, Status_Credit)
VALUES (15, 15, 25000.00, 4.25, SYSDATE, ADD_MONTHS(SYSDATE, 60), 'Activ');

INSERT INTO Tipuri_Conturi (ID_Tip_Cont, Denumire, Dobanda, Limita_Descoperire) VALUES (1, 'Curent', 0.00, 1000.00);
INSERT INTO Tipuri_Conturi (ID_Tip_Cont, Denumire, Dobanda, Limita_Descoperire) VALUES (2, 'Economii', 2.50, 0.00);
INSERT INTO Tipuri_Conturi (ID_Tip_Cont, Denumire, Dobanda, Limita_Descoperire) VALUES (3, 'Credit', 5.00, 5000.00);
INSERT INTO Tipuri_Conturi (ID_Tip_Cont, Denumire, Dobanda, Limita_Descoperire) VALUES (4, 'Business', 3.00, 10000.00);
INSERT INTO Tipuri_Conturi (ID_Tip_Cont, Denumire, Dobanda, Limita_Descoperire) VALUES (5, 'Investitii', 1.50, 0.00);

INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (1, 1, 1, 5000.00, SYSDATE, 'Activ', 1);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (2, 2, 2, 10000.00, SYSDATE, 'Activ', 2);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (3, 3, 3, 2000.00, SYSDATE, 'Activ', 3);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (4, 4, 4, 15000.00, SYSDATE, 'Activ', 4);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (5, 5, 5, 25000.00, SYSDATE, 'Activ', 5);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (6, 6, 1, 3000.00, SYSDATE, 'Activ', 6);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (7, 7, 2, 1000.00, SYSDATE, 'Activ', 7);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (8, 8, 3, 50000.00, SYSDATE, 'Activ', 8);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (9, 9, 4, 75000.00, SYSDATE, 'Activ', 9);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (10, 10, 5, 2000.00, SYSDATE, 'Activ', 1);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (11, 11, 1, 12000.00, SYSDATE, 'Activ', 2);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (12, 12, 2, 8000.00, SYSDATE, 'Activ', 3);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (13, 13, 3, 5000.00, SYSDATE, 'Activ', 4);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)

```

```

VALUES (14, 14, 4, 15000.00, SYSDATE, 'Activ', 5);
INSERT INTO Conturi (ID_Cont, ID_Client, ID_Tip_Cont, Sold, Data_Deschiderii, Stare, ID_Sucursala)
VALUES (15, 15, 5, 100000.00, SYSDATE, 'Activ', 6);

INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (1, 1, 5000.00, 2.50, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (2, 2, 10000.00, 3.00, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (3, 3, 15000.00, 2.75, SYSDATE, ADD_MONTHS(SYSDATE, 18), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (4, 4, 20000.00, 3.25, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (5, 5, 25000.00, 2.50, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (6, 6, 8000.00, 3.00, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (7, 7, 10000.00, 2.50, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (8, 8, 15000.00, 3.25, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (9, 9, 20000.00, 2.75, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (10, 10, 12000.00, 3.50, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (11, 11, 5000.00, 2.50, SYSDATE, ADD_MONTHS(SYSDATE, 12), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (12, 12, 15000.00, 3.00, SYSDATE, ADD_MONTHS(SYSDATE, 24), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (13, 13, 8000.00, 2.75, SYSDATE, ADD_MONTHS(SYSDATE, 18), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (14, 14, 20000.00, 3.25, SYSDATE, ADD_MONTHS(SYSDATE, 36), 'Activ');
INSERT INTO Depozite (ID_Depozit, ID_Cont, Suma_Depozit, Rata_Dobanzii, Data_Depunere, Data_Scadenta, Status_Depozit)
VALUES (15, 15, 30000.00, 3.50, SYSDATE, ADD_MONTHS(SYSDATE, 60), 'Activ');

INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (1, '1234567890123456', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 1);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (2, '2234567890123456', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 5000.00, 2);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (3, '3234567890123456', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 3);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (4, '4234567890123456', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 10000.00, 4);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (5, '5234567890123456', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 5);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (6, '6234567890123456', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 15000.00, 6);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (7, '7234567890123456', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 7);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (8, '8234567890123456', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 20000.00, 8);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (9, '9234567890123456', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 9);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (10, '1234567890987654', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 50000.00, 10);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (11, '2234567890987654', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 11);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (12, '3234567890987654', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 3000.00, 12);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (13, '4234567890987654', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 13);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (14, '5234567890987654', 'Credit', SYSDATE, ADD_MONTHS(SYSDATE, 60), 100000.00, 14);
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont)
VALUES (15, '6234567890987654', 'Debit', SYSDATE, ADD_MONTHS(SYSDATE, 60), NULL, 15);

```

```

INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (1, 1, 2, 500.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (2, 2, 3, 1000.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (3, 3, 4, 1500.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (4, 4, 5, 200.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (5, 5, 6, 300.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (6, NULL, 7, 1000.00, SYSDATE, 'Depunere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (7, 8, NULL, 500.00, SYSDATE, 'Retragere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (8, 9, 10, 800.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (9, 10, 1, 1200.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (10, 1, 3, 1500.00, SYSDATE, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (101, 1, 2, 500.00, SYSDATE - 10, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (102, 3, 4, 2000.00, SYSDATE - 8, 'Depunere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (103, 2, 5, 150.00, SYSDATE - 6, 'Retragere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (104, 4, 1, 800.00, SYSDATE - 15, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (105, 6, 3, 300.00, SYSDATE - 12, 'Retragere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (106, 1, 6, 1200.00, SYSDATE - 7, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (107, 5, 2, 400.00, SYSDATE - 14, 'Depunere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (108, 2, 4, 250.00, SYSDATE - 5, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (109, 3, 6, 600.00, SYSDATE - 9, 'Retragere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (110, 6, 5, 100.00, SYSDATE - 4, 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (201, 1, 2, 500.00, TO_DATE('2025-01-07 08:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (202, 3, 4, 1200.00, TO_DATE('2025-01-07 09:15:00', 'YYYY-MM-DD HH24:MI:SS'), 'Depunere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (203, 2, 5, 800.00, TO_DATE('2025-01-07 11:45:00', 'YYYY-MM-DD HH24:MI:SS'), 'Retragere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (204, 4, 1, 1500.00, TO_DATE('2025-01-07 18:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (205, 5, 3, 600.00, TO_DATE('2025-01-07 20:15:00', 'YYYY-MM-DD HH24:MI:SS'), 'Depunere');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (206, 3, 1, 700.00, TO_DATE('2025-01-07 23:45:00', 'YYYY-MM-DD HH24:MI:SS'), 'Transfer');
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (207, 4, 2, 400.00, TO_DATE('2025-01-07 04:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'Retragere');

```

Alte operații LDD:

1) Să se adauge o restricție de integritate la coloana Tip_Tranzactie din tabela Tranzactii astfel încât să accepte doar valorile Transfer, Depunere și Retragere. Înregistrările existente care nu respectă această condiție vor fi șterse înainte de aplicarea restricției.

DELETE FROM Tranzactii

WHERE Tip_Tranzactie NOT IN ('Transfer', 'Depunere', 'Retragere');

ALTER TABLE Tranzactii

ADD CONSTRAINT check_Tip_Tranzactie

CHECK (Tip_Tranzactie IN ('Transfer', 'Depunere', 'Retragere'));

2) Să se adauge o coloană Status în tabela Tranzactii, care să accepte doar valorile Reusit și Nereusit, cu valoarea implicită Reusit.

ALTER TABLE Tranzactii

ADD Status VARCHAR2(10) DEFAULT 'Reusit' CHECK (Status IN ('Reusit', 'Nereusit'));

3) Să se eliminate tabela Credite din baza de date

DROP TABLE Credite CASCADE CONSTRAINTS;

DESCRIBE Credite;

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, there are tabs for 'Welcome Page' and 'SerbanSilviu_1065'. Below the toolbar, the 'Worksheet' tab is selected. The main workspace contains the following SQL code:

```
DROP TABLE Credite CASCADE CONSTRAINTS;
DESCRIBE Credite;
```

In the 'Script Output' pane at the bottom, the results of the execution are shown:

```
Table CREDITE dropped.

ERROR:
ORA-04043: object Credite does not exist
```

4) Să se recupereze tabela Credite stearsa anterior

FLASHBACK TABLE Credite TO BEFORE DROP;

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, there are tabs for 'Welcome Page', 'SerbanSilviu_1065', and 'CREDITE'. Below the tabs, there are icons for running, saving, and zooming. The main area has two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab contains the following SQL code:

```
FLASHBACK TABLE Credite TO BEFORE DROP;
DESCRIBE Credite;
```

Below the code, the 'Script Output' tab shows the results of the execution:

```
Table CREDITE dropped.

ERROR:
ORA-04043: object Credite does not exist

Flashback succeeded.

Name      Null?    Type
-----  -----
ID_CREDIT  NOT NULL NUMBER(10)
ID_CLIENT   NOT NULL NUMBER(10)
SUMA_APROBATA  NUMBER(12,2)
RATA_DOBANZII  NUMBER(5,2)
DATA_ACORDARII DATE
DATA_SCADENTEI DATE
STATUS_CREDIT  VARCHAR2(20)
```

5) Să se adauge o coloană Status în tabela **Carduri**, care să accepte doar valorile Valabil și Expirat, având implicit valoarea Valabil pentru toate noile înregistrări și înregistrările existente.

ALTER TABLE Carduri

ADD Status VARCHAR2(10) DEFAULT 'Valabil' CHECK (Status IN ('Valabil', 'Expirat')) NOT NULL;

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, there are tabs for 'Welcome Page', 'SerbanSilviu_1065', and 'CREDITE'. Below the tabs, there are icons for running, saving, and zooming. The main area has two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab contains the following SQL code:

```
ALTER TABLE Carduri
ADD Status VARCHAR2(10) DEFAULT 'Valabil' CHECK (Status IN ('Valabil', 'Expirat')) NOT NULL;
```

Below the code, the 'Script Output' tab shows the results of the execution:

```
Table CARDURI altered.
```

- 6) Să se adauge o coloană nouă denumită ManagerId în tabela **Angajați**, care să permită crearea unei structuri ierarhice între angajați.

```
ALTER TABLE Angajati
```

```
ADD ManagerId NUMBER;
```

```
DESCRIBE Angajati;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are two tabs: 'SerbanSilviu_1065.sql' and 'SerbanSilviu_1065'. The 'Query Builder' tab is active, displaying the following SQL code:

```
ALTER TABLE Angajati
ADD ManagerId NUMBER;
DESCRIBE Angajati;
```

In the bottom-left pane, the 'Script Output' tab is active, showing the results of the executed query. It displays the message 'Table ANGAJATI altered.' followed by the table structure:

Name	Null?	Type
ID_ANGAJAT	NOT NULL	NUMBER(10)
PRENUME	NOT NULL	VARCHAR2(50)
NUME	NOT NULL	VARCHAR2(50)
CNP	NOT NULL	VARCHAR2(13)
ID_SUCURSALA	NOT NULL	NUMBER(10)
POZITIE	NOT NULL	VARCHAR2(30)
DATA_ANGAJARII		DATE
SALARIU	NOT NULL	NUMBER(10,2)
TELEFON		VARCHAR2(15)
EMAIL		VARCHAR2(50)
MANAGERID		NUMBER

A status bar at the bottom of the 'Script Output' tab indicates 'Task completed in 2.803 seconds'.

Alte operații LMD:

1) Să se populeze coloana ManagerId astfel:

- **Directorul General** (ID_Angajat = 0): ManagerId = NULL,
- **Managerii sucursalelor**: ManagerId = 0,
- **Casierii și Consultantii**: ManagerId = ID-ul managerului din aceeași sucursală.

```
UPDATE Angajati
SET ManagerId = NULL
WHERE ID_Angajat = 0;
UPDATE Angajati
SET ManagerId = 0
WHERE Pozitie = 'Manager';
UPDATE Angajati
SET ManagerId = (
    SELECT ID_Angajat
    FROM Angajati Manager
    WHERE Manager.Pozitie = 'Manager'
        AND Manager.ID_Sucursala = Angajati.ID_Sucursala
)
WHERE Pozitie IN ('Casier', 'Consultant Financiar');
```

The screenshot shows a database interface with two main panes. The top pane is titled 'Worksheet' and contains the SQL code for updating the 'Angajati' table. The bottom pane is titled 'Script Output' and displays the results of the executed query, showing three rows updated: 1 row updated, 10 rows updated, and 20 rows updated.

```
Worksheet Query Builder
UPDATE Angajati
SET ManagerId = NULL
WHERE ID_Angajat = 0;
UPDATE Angajati
SET ManagerId = 0
WHERE Pozitie = 'Manager';
UPDATE Angajati
SET ManagerId = (
    SELECT ID_Angajat
    FROM Angajati Manager
    WHERE Manager.Pozitie = 'Manager'
        AND Manager.ID_Sucursala = Angajati.ID_Sucursala
)
WHERE Pozitie IN ('Casier', 'Consultant Financiar');

Script Output x
redo | Task completed in 0.155 seconds
1 row updated.

10 rows updated.

20 rows updated.
```

Columns Data Model | Constraints Grants | Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL Actions...

ID_ANGAJAT	PRENUME	NUME	CNP	ID_SUCURSALA	POZITIE	DATA_ANGAJARII	SALARIU	TELEFON	EMAIL	MANAGERID
1	O Silviu	Serban	1900101123456	1	Director	08-JAN-25	10000	0745123456	silviu.serban@gmail.com	(null)
2	Ion	Popescu	5900101123456	1	Manager	08-JAN-25	5000	0740123456	ion.popescu@gmail.com	0
3	2 Maria	Ionescu	2910101123456	1	Casier	08-JAN-25	3000	0741123456	maria.ionescu@yahoo.com	1
4	3 Andrei	Georgescu	1920101123456	1	Consultant Financiar	08-JAN-25	3500	0742123456	andrei.georgescu@gmail.com	1
5	Ana	Dumitrescu	2930101123456	2	Manager	08-JAN-25	5000	0743123456	ana.dumitrescu@yahoo.com	0
6	5 Mihai	Vasilescu	1940101123456	2	Casier	08-JAN-25	3000	0744123456	mihai.vasilescu@gmail.com	4
7	6 Elena	Popa	2950101123456	2	Consultant Financiar	08-JAN-25	3500	0745123456	elena.popa@yahoo.com	4
8	7 Cristian	Stan	1960101123456	3	Manager	08-JAN-25	5000	0746123456	cristian.stan@gmail.com	0
9	8 Alina	Marinescu	2970101123456	3	Casier	08-JAN-25	3000	0747123456	alina.marinescu@yahoo.com	7
10	9 George	Ilieșcu	1980101123456	3	Consultant Financiar	08-JAN-25	3500	0748123456	george.ilieșcu@gmail.com	7
11	10 Raluca	Stanciu	2990101123456	4	Manager	08-JAN-25	5000	0749123456	raluca.stanciu@yahoo.com	0
12	11 Alex	Moldoveanu	2000101123456	4	Casier	08-JAN-25	3000	0750123456	alex.moldoveanu@gmail.com	10
13	12 Diana	Tudor	2010101123456	4	Consultant Financiar	08-JAN-25	3500	0751123456	diana.tudor@yahoo.com	10
14	13 Sorin	Radulescu	2020101123456	5	Manager	08-JAN-25	5000	0752123456	sorin.radulescu@gmail.com	0
15	14 Simona	Calinescu	2030101123456	5	Casier	08-JAN-25	3000	0753123456	simona.calinescu@yahoo.com	13
16	15 Victor	Grigorescu	2040101123456	5	Consultant Financiar	08-JAN-25	3500	0754123456	victor.grigorescu@gmail.com	13
17	16 Adrian	Popa	2050101123456	6	Manager	08-JAN-25	5000	0755123456	adrian.popa@mail.com	0
18	17 Gabriel	Marin	2060101123456	6	Casier	08-JAN-25	3000	0756123456	gabriel.marin@yahoo.com	16
19	18 Roxana	Enache	2070101123456	6	Consultant Financiar	08-JAN-25	3500	0757123456	roxana.enache@gmail.com	16
20	19 Oana	Mihai	2080101123456	7	Manager	08-JAN-25	5000	0758123456	oana.mihai@yahoo.com	0
21	20 Vlad	Tudor	2090101123456	7	Casier	08-JAN-25	3000	0759123456	vlad.tudor@gmail.com	19
22	21 Iulia	Dobre	2100101123456	7	Consultant Financiar	08-JAN-25	3500	0760123456	iulia.dobre@yahoo.com	19
23	22 Marius	Popescu	2110101123456	8	Manager	08-JAN-25	5000	0761123456	marius.popescu@gmail.com	0
24	23 Carmen	Toma	2120101123456	8	Casier	08-JAN-25	3000	0762123456	carmen.toma@yahoo.com	22
25	24 Alin	Georgescu	2130101123456	8	Consultant Financiar	08-JAN-25	3500	0763123456	alin.georgescu@gmail.com	22
26	25 Dan	Iordache	2140101123456	9	Manager	08-JAN-25	5000	0764123456	dan.iordache@gmail.com	0
27	26 Roxana	Stoian	2150101123456	9	Casier	08-JAN-25	3000	0765123456	roxana.stoian@yahoo.com	25
28	27 Cristina	Dumitru	2160101123456	9	Consultant Financiar	08-JAN-25	3500	0766123456	cristina.dumitru@gmail.com	25
29	28 Alexandru	Serban	2170101123456	10	Manager	08-JAN-25	5000	0767123456	alexandru.serban@gmail.com	0
30	29 Bianca	Enescu	2180101123456	10	Casier	08-JAN-25	3000	0768123456	bianca.enescu@yahoo.com	28
31	30 Razvan	Dinu	2190101123456	10	Consultant Financiar	08-JAN-25	3500	0769123456	razvan.dinu@gmail.com	28

2) Să se actualizeze tabela **Depozite**, astfel încât rata dobânzii să fie redusă cu **0.5** pentru toate înregistrările în care suma depozitată este mai mare de **10.000**.

UPDATE Depozite

```
SET Rata_Dobanzii = Rata_Dobanzii - 0.5
WHERE Suma_Depozit > 10000;
SELECT * FROM Depozite WHERE Suma_Depozit > 10000;
```

Script Output | Query Result | All Rows Fetched: 8 in 0.023 seconds

ID_DEPOZIT	ID_CONT	SUMA_DEPOZIT	RATA_DOBANZII	DATA_DEPUNERE	DATA_SCADENTA	STATUS_DEPOZIT
1	3	3	15000	2.25 07-JAN-25	07-JUL-26	Activ
2	4	4	20000	2.75 07-JAN-25	07-JAN-28	Activ
3	8	8	15000	2.75 07-JAN-25	07-JAN-26	Activ
4	9	9	20000	2.25 07-JAN-25	07-JAN-27	Activ
5	10	10	12000	3 07-JAN-25	07-JAN-28	Activ
6	12	12	15000	2.5 07-JAN-25	07-JAN-27	Activ
7	14	14	20000	2.75 07-JAN-25	07-JAN-28	Activ
8	15	15	30000	3 07-JAN-25	07-JAN-30	Activ

3) Să se adauge un nou client în tabela **Clienți** cu valorile necesare pentru toate coloanele.

```
INSERT INTO Clienti (ID_Client, Nume, Prenume, CNP, Adresa, Telefon, Email, Data_Inregistrarii)
VALUES (16, 'Popescu', 'Andrei', '1990501123456', 'Strada Libertății 15', '0745678901',
'andrei.popescu@gmail.com', SYSDATE);
```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL code for inserting a new client record. Below the worksheet, the 'Script Output' tab shows the result: '1 row inserted.'

4) Să se șteargă din tabela **Clienți** înregistrarea care corespunde clientului cu **ID_Client = 16**.

```
DELETE FROM Clienti
WHERE ID_Client = 16;
```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL code for deleting a client record where ID_Client = 16. Below the worksheet, the 'Script Output' tab shows the result: '1 row deleted.'

5) Să se realizeze un transfer de **100 de lei** între contul cu **ID_Cont = 6** (sursa) și contul cu **ID_Cont = 11** (destinație), iar tranzacția să fie înregistrată în tabela **Tranzactii** cu o dată din anul **2022** (ex: 15 decembrie 2022).

```
UPDATE Conturi
SET Sold = Sold - 100
WHERE ID_Cont = 6;
UPDATE Conturi
SET Sold = Sold + 100
WHERE ID_Cont = 11;
```

```
INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma,
Data_Tranzactiei, Tip_Tranzactie)
VALUES (20, 6, 11, 100.00, TO_DATE('2022-12-15', 'YYYY-MM-DD'), 'Transfer');
```

The screenshot shows the SSMS interface. In the top window (Worksheet), there is a code editor containing the following SQL script:

```

UPDATE Conturi
SET Sold = Sold - 100
WHERE ID_Cont = 6;
UPDATE Conturi
SET Sold = Sold + 100
WHERE ID_Cont = 11;

INSERT INTO Tranzactii (ID_Tranzactie, ID_Cont_Sursa, ID_Cont_Destinatie, Suma, Data_Tranzactiei, Tip_Tranzactie)
VALUES (20, 6, 11, 100.00, TO_DATE('2022-12-15', 'YYYY-MM-DD'), 'Transfer');

```

In the bottom window (Script Output), the results of the execution are displayed:

```

1 row updated.

1 row updated.

1 row inserted.

```

6) Ștergere tranzacțiile mai vechi de un an.

`DELETE FROM Tranzactii`

`WHERE Data_Tranzactiei < ADD_MONTHS(SYSDATE, -12);`

The screenshot shows the SSMS interface. In the top window (Worksheet), there is a code editor containing the following SQL script:

```

DELETE FROM Tranzactii
WHERE Data_Tranzactiei < ADD_MONTHS(SYSDATE, -12);

```

In the bottom window (Script Output), the results of the execution are displayed:

```

1 row deleted.

```

7) Să se mărească salariul cu **10%** pentru toți angajații care au poziția **Casier** în tabela **Angajați**.

UPDATE Angajati

SET Salariu = Salariu * 1.10

WHERE Pozitie = 'Casier';

The screenshot shows the SQL Server Management Studio interface. The title bar indicates the connection is to 'SerbanSilviu_1065' under the database 'ANGAJATI'. The main window displays a 'Worksheet' tab with the following T-SQL code:

```
UPDATE Angajati  
SET Salariu = Salariu * 1.10  
WHERE Pozitie = 'Casier';
```

Below the worksheet, the 'Script Output' pane shows the message 'Task completed in 0.029 seconds'. The status bar at the bottom of the interface displays '10 rows updated.'

8) Să se acorde suma de **50 de lei** fiecărui client din tabela **Conturi** care are numele sau prenumele Ion sau Ioana în tabela **Clienți**.

UPDATE Conturi

SET Sold = Sold + 50

WHERE ID_Client IN (

SELECT ID_Client

FROM Clienti

WHERE Nume IN ('Ion', 'Ioana') OR Prenume IN ('Ion', 'Ioana')

);

The screenshot shows the SQL Server Management Studio interface. The title bar indicates the connection is to 'SerbanSilviu_1065' under the database 'CLIENTI'. The main window displays a 'Worksheet' tab with the following T-SQL code:

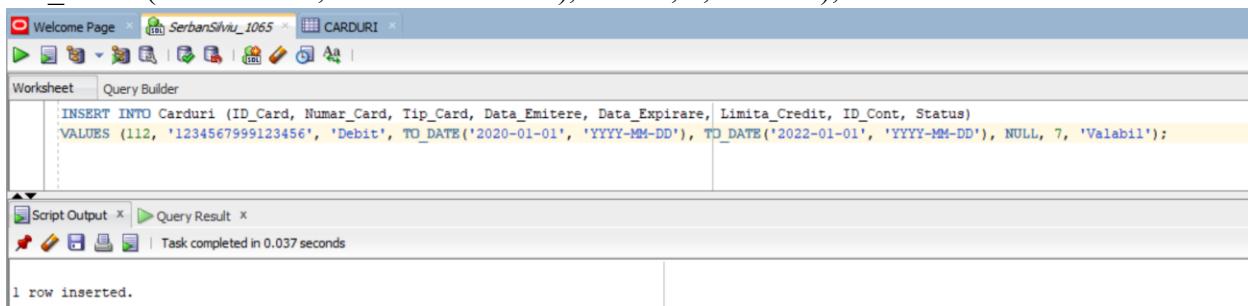
```
UPDATE Conturi  
SET Sold = Sold + 50  
WHERE ID_Client IN (  
    SELECT ID_Client  
    FROM Clienti  
    WHERE Nume IN ('Ion', 'Ioana') OR Prenume IN ('Ion', 'Ioana')  
)
```

Below the worksheet, the 'Script Output' pane shows the message 'Task completed in 0.04 seconds'. The status bar at the bottom of the interface displays '1 row updated.'

9) Să se adauge un card expirat în tabela **Carduri** pentru clientul cu **ID_Client = 7**, astfel încât data de expirare să fie în trecut, iar statusul cardului să fie Valabil.

```
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont, Status)
```

```
VALUES (112, '1234567999123456', 'Debit', TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('2022-01-01', 'YYYY-MM-DD'), NULL, 7, 'Valabil');
```



The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'Welcome Page', 'SerbanSilviu_1065', and 'CARDURI'. The main area is titled 'Worksheet' and contains the following SQL code:

```
INSERT INTO Carduri (ID_Card, Numar_Card, Tip_Card, Data_Emitere, Data_Expirare, Limita_Credit, ID_Cont, Status)
VALUES (112, '1234567999123456', 'Debit', TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('2022-01-01', 'YYYY-MM-DD'), NULL, 7, 'Valabil');
```

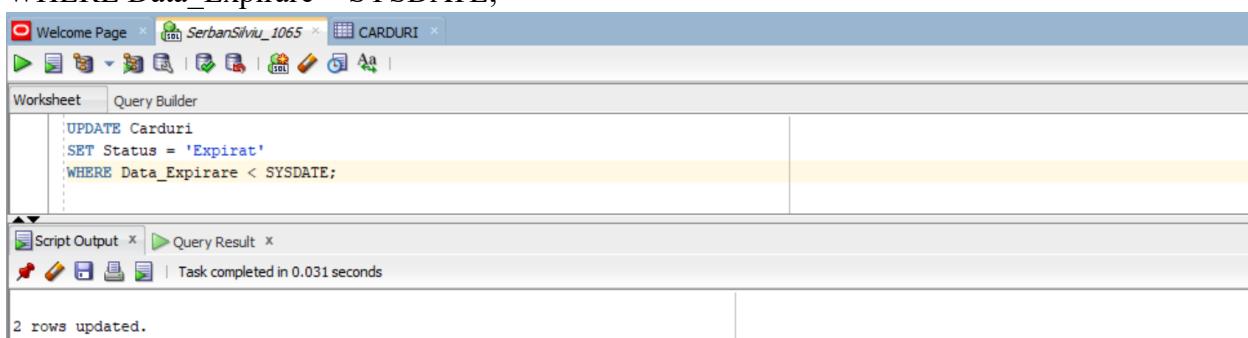
Below the code, the status bar indicates 'Task completed in 0.037 seconds'. At the bottom, it says '1 row inserted.'

10) Să se actualizeze coloana Status din tabela **Carduri** pentru toate cardurile a căror dată de expirare (Data_Expirare) este anterioară datei curente, astfel încât Status să fie setat la Expirat.

```
UPDATE Carduri
```

```
SET Status = 'Expirat'
```

```
WHERE Data_Expirare < SYSDATE;
```



The screenshot shows the Oracle SQL Developer interface. The top bar has tabs for 'Welcome Page', 'SerbanSilviu_1065', and 'CARDURI'. The main area is titled 'Worksheet' and contains the following SQL code:

```
UPDATE Carduri
SET Status = 'Expirat'
WHERE Data_Expirare < SYSDATE;
```

Below the code, the status bar indicates 'Task completed in 0.031 seconds'. At the bottom, it says '2 rows updated.'

Exemple de interogări

- 1) Să se afișeze angajații care lucrează în sucursalele din **Timișoara**, **Cluj**, sau **București**, incluzând ID-ul angajatului, prenumele, numele și numele sucursalei.

```
SELECT a.ID_Angajat, a.Prenume, a.Nume, s.Nume_Sucursala  
FROM Angajati a  
JOIN Sucursala s ON a.ID_Sucursala = s.ID_Sucursala  
WHERE s.Nume_Sucursala IN ('Timisoara', 'Cluj', 'Bucuresti');
```

The screenshot shows the Oracle SQL Developer interface. The top window is titled 'Worksheet' and contains the SQL query. The bottom window is titled 'Query Result' and displays the results of the query execution. The results are presented in a table with four columns: ID_ANGAJAT, PRENUME, NUME, and NUME_SUCURSALA. The data is as follows:

ID_ANGAJAT	PRENUME	NUME	NUME_SUCURSALA
1	Silviu	Serban	Bucuresti
2	Maria	Ionescu	Bucuresti
3	Andrei	Georgescu	Bucuresti
4	Ana	Dumitrescu	Cluj
5	Mihai	Vasilescu	Cluj
6	Elena	Popa	Cluj
7	Cristian	Stan	Timisoara
8	Alina	Marinescu	Timisoara
9	George	Iliescu	Timisoara

- 2) Să se afișeze clienții cu vârsta cuprinsă între **18 și 35 de ani**, calculată pe baza **CNP-ului**, incluzând ID-ul clientului, numele, prenumele și vârsta lor.

```
SELECT ID_Client, Nume, Prenume, CNP,  
       EXTRACT(YEAR FROM SYSDATE) - CASE  
         WHEN SUBSTR(CNP, 1, 1) IN ('1', '2') THEN 1900 + TO_NUMBER(SUBSTR(CNP, 2, 2))  
         WHEN SUBSTR(CNP, 1, 1) IN ('5', '6') THEN 2000 + TO_NUMBER(SUBSTR(CNP, 2, 2))  
       END AS Varsta  
FROM Clienti  
WHERE EXTRACT(YEAR FROM SYSDATE) - CASE  
         WHEN SUBSTR(CNP, 1, 1) IN ('1', '2') THEN 1900 + TO_NUMBER(SUBSTR(CNP, 2, 2))  
         WHEN SUBSTR(CNP, 1, 1) IN ('5', '6') THEN 2000 + TO_NUMBER(SUBSTR(CNP, 2, 2))  
       END BETWEEN 18 AND 35;
```

SQL Worksheet

```

SELECT ID_Client, Nume, Prenume, CNP,
       EXTRACT(YEAR FROM SYSDATE) - CASE
           WHEN SUBSTR(CNP, 1, 1) IN ('1', '2') THEN 1900 + TO_NUMBER(SUBSTR(CNP, 2, 2))
           WHEN SUBSTR(CNP, 1, 1) IN ('5', '6') THEN 2000 + TO_NUMBER(SUBSTR(CNP, 2, 2))
       END AS Varsta
FROM Clienti
WHERE EXTRACT(YEAR FROM SYSDATE) - CASE
           WHEN SUBSTR(CNP, 1, 1) IN ('1', '2') THEN 1900 + TO_NUMBER(SUBSTR(CNP, 2, 2))
           WHEN SUBSTR(CNP, 1, 1) IN ('5', '6') THEN 2000 + TO_NUMBER(SUBSTR(CNP, 2, 2))
       END BETWEEN 18 AND 35;

```

Script Output | Query Result

ID_CLIENT	NUME	PRENUME	CNP	VARSTA
1	Popescu	Ion	1900101123456	35
2	Ionescu	Maria	2910101123456	34
3	Georgescu	Andrei	1920101123456	33
4	Dumitrescu	Ana	2930101123456	32
5	Vasilescu	Mihai	1940101123456	31
6	Popa	Elena	2950101123456	30
7	Stan	Cristian	1960101123456	29
8	Marinescu	Alina	2970101123456	28
9	Iliescu	George	1980101123456	27
10	Stanciu	Raluca	2990101123456	26

3) Să se afișeze clienții care au soldul total în toate conturile mai mare de **10.000**, incluzând ID-ul clientului, numele, prenumele și suma totală a soldurilor lor.

```

SELECT c.ID_Client, cl.Nume, cl.Prenume, SUM(c.Sold) AS Sold_Total
FROM Conturi c

```

```

JOIN Clienti cl ON c.ID_Client = cl.ID_Client
GROUP BY c.ID_Client, cl.Nume, cl.Prenume
HAVING SUM(c.Sold) > 10000;

```

SQL Worksheet

```

SELECT c.ID_Client, cl.Nume, cl.Prenume, SUM(c.Sold) AS Sold_Total
FROM Conturi c
JOIN Clienti cl ON c.ID_Client = cl.ID_Client
GROUP BY c.ID_Client, cl.Nume, cl.Prenume
HAVING SUM(c.Sold) > 10000;

```

Script Output | Query Result

ID_CLIENT	NUME	PRENUME	SOLD_TOTAL
1	Dumitrescu	Ana	15000
2	Vasilescu	Mihai	25000
3	Marinescu	Alina	50000
4	Iliescu	George	75000
5	Moldoveanu	Alex	12300
6	Calinescu	Simona	15000
7	Grigorescu	Victor	100000

4) Să se afișeze toate conturile de tip **Curent** care au fost deschise în sucursalele din **București** și **Cluj**, incluzând ID-ul contului, ID-ul clientului, soldul, data deschiderii și numele sucursalei.

SELECT ID_Cont, ID_Client, Sold, Data_Deschiderii, ID_Sucursala

FROM Conturi

WHERE ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Curent')

AND ID_Sucursala IN (SELECT ID_Sucursala FROM Sucursala WHERE Nume_Sucursala IN

('Bucuresti', 'Cluj'));

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the query code for question 4. Below the worksheet, the 'Query Result' tab is open, showing a table with two rows of data. The columns are labeled: ID_CONT, ID_CLIENT, SOLD, DATA_DESCHIDERII, and ID_SUCURSALA. The data is as follows:

ID_CONT	ID_CLIENT	SOLD	DATA_DESCHIDERII	ID_SUCURSALA
1	1	1	5050 07-JAN-25	1
2	11	11	12300 07-JAN-25	2

5) Să se acorde un bonus de **50 de lei** tuturor clienților care își serbează astăzi ziua de naștere, actualizând soldurile conturilor acestora. Afisarea să includă doar ID-ul clientului, luna și ziua nașterii extrase din **CNP**.

UPDATE Conturi

SET Sold = Sold + 50

WHERE ID_Client IN (

SELECT ID_Client

FROM Clienti

WHERE TO_CHAR(SYSDATE, 'MMDD') = TO_CHAR(TO_DATE(SUBSTR(CNP, 2, 6), 'YYMMDD'), 'MMDD')

);

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the query code for question 5. Below the worksheet, the 'Query Result' tab is open, showing a message: '0 rows updated.' This indicates that the update did not affect any rows because the WHERE clause was not satisfied.

6) Să se afișeze clienții care au conturi cu solduri mai mari decât media tuturor conturilor, incluzând informații despre client (ID, nume, prenume) și cont (ID, sold).

SELECT ID_Client, Nume, Prenume FROM Clienti WHERE ID_Client IN (SELECT ID_Client FROM Conturi WHERE Sold > (SELECT AVG(Sold) FROM Conturi));

The screenshot shows the SQL Server Management Studio interface. The top bar has tabs for 'Welcome Page' and 'SerbanSilviu_1065'. Below the bar are standard toolbar icons. The main area is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the following SQL code:

```

SELECT ID_Client, Nume, Prenume
FROM Clienti
WHERE ID_Client IN (
    SELECT ID_Client
    FROM Conturi
    WHERE Sold > (SELECT AVG(Sold) FROM Conturi)
);

```

The 'Query Result' tab is selected at the bottom, showing the results of the query:

ID_CLIENT	NUME	PRENUME
1	5	Vasilescu Mihai
2	8	Marinescu Alina
3	9	Ilieșcu George
4	15	Grigorescu Victor

Execution details at the bottom indicate: All Rows Fetched: 4 in 0.088 seconds.

7) Să se afișeze toate conturile de tip **Curent** care au mai puțin de **3 tranzacții**, incluzând ID-ul contului și numărul total de tranzacții.

SELECT c.ID_Cont, c.Sold, COUNT(t.ID_Tranzactie) AS Numar_Tranzactii
FROM Conturi c

LEFT JOIN Tranzactii t ON c.ID_Cont = t.ID_Cont_Sursa

WHERE c.ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Curent')
GROUP BY c.ID_Cont, c.Sold

HAVING COUNT(t.ID_Tranzactie) < 3;

The screenshot shows the SQL Server Management Studio interface. The top bar has tabs for 'Welcome Page' and 'SerbanSilviu_1065'. Below the bar are standard toolbar icons. The main area is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the following SQL code:

```

SELECT c.ID_Cont, c.Sold, COUNT(t.ID_Tranzactie) AS Numar_Tranzactii
FROM Conturi c
LEFT JOIN Tranzactii t ON c.ID_Cont = t.ID_Cont_Sursa
WHERE c.ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Curent')
GROUP BY c.ID_Cont, c.Sold
HAVING COUNT(t.ID_Tranzactie) < 3;

```

The 'Query Result' tab is selected at the bottom, showing the results of the query:

ID_CONT	SOLD	NUMAR_TRANZACTII
1	1 5050	2
2	6 2700	0
3	11 12300	0

Execution details at the bottom indicate: All Rows Fetched: 3 in 0.034 seconds.

8) Să se afișeze sucursalele ordonate descrescător după suma totală a depozitelor clientilor.

```
SELECT s.Nume_Sucursala, SUM(d.Suma_Depozit) AS Total_Depozite  
FROM Sucursala s  
JOIN Conturi c ON s.ID_Sucursala = c.ID_Sucursala  
JOIN Depozite d ON c.ID_Cont = d.ID_Cont  
GROUP BY s.Nume_Sucursala  
ORDER BY Total_Depozite DESC;
```

The screenshot shows the SQL Server Management Studio interface. The top bar has tabs for 'Welcome Page', 'SerbanSilviu_1065', 'DEPOZITE', and 'Relational_1 (Untitled_1)'. Below the tabs is a toolbar with icons for New Query, Open, Save, Print, Copy, Paste, Find, Replace, and others. The main area is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the SQL query from above. The 'Query Result' pane shows the output in a table:

NUME_SUCURSALA	TOTAL_DEPOZITE
1 Brasov	38000
2 Timisoara	30000
3 Iasi	28000
4 Constanta	22500
5 Arad	20000
6 Bucuresti	17000
7 Craiova	15000
8 Cluj	15000
9 Sibiu	10000

9) Să se afișeze conturile care nu au fost implicate niciodată în nicio tranzacție.

```
SELECT ID_Cont  
FROM Conturi  
MINUS  
SELECT ID_Cont_Sursa  
FROM Tranzactii  
UNION  
SELECT ID_Cont_Destinatie  
FROM Tranzactii;
```

The screenshot shows the SQL Server Management Studio interface. The top bar has tabs for 'Welcome Page', 'SerbanSilviu_1065', 'DEPOZITE', and 'Relational_1 (Untitled_1)'. Below the tabs is a toolbar with icons for New Query, Open, Save, Print, Copy, Paste, Find, Replace, and others. The main area is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the SQL query from above. The 'Query Result' pane shows the output in a table:

ID_CONT
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

10) Să se modifice tipul conturilor de **Economii** în conturi de **Curent** și să se actualizeze soldul acestora prin adăugarea dobânzii aferente.

UPDATE Conturi c

```
SET ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Curent'),
```

```
    Sold = Sold + (Sold * (SELECT Dobanda / 100 FROM Tipuri_Conturi WHERE ID_Tip_Cont = c.ID_Tip_Cont))
```

```
WHERE ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Economii');
```

```
SELECT c.ID_Cont, c.ID_Client, tc.Denumire AS Tip_Cont, c.Sold
```

```
FROM Conturi c
```

```
JOIN Tipuri_Conturi tc ON c.ID_Tip_Cont = tc.ID_Tip_Cont
```

```
WHERE tc.Denumire IN ('Curent');
```

```
UPDATE Conturi c
SET ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Curent'),
    Sold = Sold + (Sold * (SELECT Dobanda / 100 FROM Tipuri_Conturi WHERE ID_Tip_Cont = c.ID_Tip_Cont))
WHERE ID_Tip_Cont = (SELECT ID_Tip_Cont FROM Tipuri_Conturi WHERE Denumire = 'Economii');

SELECT c.ID_Cont, c.ID_Client, tc.Denumire AS Tip_Cont, c.Sold
FROM Conturi c
JOIN Tipuri_Conturi tc ON c.ID_Tip_Cont = tc.ID_Tip_Cont
WHERE tc.Denumire IN ('Curent');
```

Script Output | Query Result | Task completed in 0.151 seconds

3 rows updated.

ID_CONT	ID_CLIENT	TIP_CONT	SOLD
1	1	Curent	5050
2	2	Curent	10250
6	6	Curent	2700
7	7	Curent	1025
11	11	Curent	12300
12	12	Curent	8200

6 rows selected.

11) Să se realizeze o clasificare a clientilor în funcție de soldul total al acestora, folosind CASE pentru a defini categoriile (Sold Mic, Sold Mediu, Sold Mare). Rezultatul să includă ID-ul clientului, numele, prenumele, soldul total și categoria.

```
SELECT cl.ID_Client, cl.Nume, cl.Prenume, SUM(c.Sold) AS Sold_Total,
```

```
CASE
```

```
WHEN SUM(c.Sold) < 1000 THEN 'Sold Mic'
```

```
WHEN SUM(c.Sold) BETWEEN 1000 AND 10000 THEN 'Sold Mediu'
```

```
WHEN SUM(c.Sold) > 10000 THEN 'Sold Mare'
```

```
ELSE 'Neclasificat'
```

```
END AS Categorie
```

```
FROM Clienti cl
```

```
JOIN Conturi c ON cl.ID_Client = c.ID_Client
```

```
GROUP BY cl.ID_Client, cl.Nume, cl.Prenume;
```

Worksheet | Query Builder

```

SELECT cl.ID_Client, cl.Nume, cl.Prenume, SUM(c.Sold) AS Sold_Total,
CASE
    WHEN SUM(c.Sold) < 1000 THEN 'Sold Mic'
    WHEN SUM(c.Sold) BETWEEN 1000 AND 10000 THEN 'Sold Mediu'
    WHEN SUM(c.Sold) > 10000 THEN 'Sold Mare'
    ELSE 'Neclasificat'
END AS Categorie
FROM Clienti cl
JOIN Conturi c ON cl.ID_Client = c.ID_Client
GROUP BY cl.ID_Client, cl.Nume, cl.Prenume;

```

Script Output | Query Result | SQL | All Rows Fetched: 15 in 0.031 seconds

ID_CLIENT	NUME	PRENUME	SOLD_TOTAL	CATEGORIE
1	Popescu	Ion	5050	Sold Mediu
2	Ionescu	Maria	10250	Sold Mare
3	Georgescu	Andrei	2000	Sold Mediu
4	Dumitrescu	Ana	15000	Sold Mare
5	Vasilescu	Mihai	25000	Sold Mare
6	Popa	Elena	2700	Sold Mediu
7	Stan	Cristian	1025	Sold Mediu
8	Marinescu	Alina	50000	Sold Mare
9	Ilieșcu	George	75000	Sold Mare
10	Stanciu	Raluca	2000	Sold Mediu
11	Moldoveanu	Alex	12300	Sold Mare
12	Tudor	Diana	8200	Sold Mediu
13	Radulescu	Sorin	5000	Sold Mediu
14	Calinescu	Simona	15000	Sold Mare
15	Grigorescu	Victor	100000	Sold Mare

12) Să se determine clientul care a realizat cele mai multe tranzacții în ultima lună, incluzând numele, prenumele și numărul total de tranzacții.

```

SELECT cl.ID_Client, cl.Nume, cl.Prenume, COUNT(t.ID_Tranzactie) AS Numar_Tranzactii
FROM Clienti cl
JOIN Conturi c ON cl.ID_Client = c.ID_Client
JOIN Tranzactii t ON c.ID_Cont = t.ID_Cont_Sursa
WHERE t.Data_Tranzactiei >= ADD_MONTHS(SYSDATE, -1)
GROUP BY cl.ID_Client, cl.Nume, cl.Prenume
ORDER BY Numar_Tranzactii DESC
FETCH FIRST 1 ROWS ONLY;

```

Worksheet | Query Builder

```

SELECT cl.ID_Client, cl.Nume, cl.Prenume, COUNT(t.ID_Tranzactie) AS Numar_Tranzactii
FROM Clienti cl
JOIN Conturi c ON cl.ID_Client = c.ID_Client
JOIN Tranzactii t ON c.ID_Cont = t.ID_Cont_Sursa
WHERE t.Data_Tranzactiei >= ADD_MONTHS(SYSDATE, -1)
GROUP BY cl.ID_Client, cl.Nume, cl.Prenume
ORDER BY Numar_Tranzactii DESC
FETCH FIRST 1 ROWS ONLY;

```

Script Output | Query Result | SQL | All Rows Fetched: 1 in 0.042 seconds

ID_CLIENT	NUME	PRENUME	NUMAR_TANZACTII
1	Popescu	Ion	4

13) Să se afișeze structura ierarhică a angajaților din baza de date, pornind de la Directorul General și continuând cu managerii, casierii și consultanții, utilizând clauza **CONNECT BY**. Rezultatul trebuie să includă nivelul ierarhic al fiecărui angajat, ID-ul, numele complet, poziția, ID-ul sucursalei, ID-ul managerului și lanțul ierarhic complet (de la director până la angajat).

```

SELECT
    LEVEL AS Nivel_Ierarhic,
    ID_Angajat,
    Prenume || ' ' || Nume AS Nume_Complet,
    Pozitie,
    ID_Sucursala,
    ManagerId,
    SYS_CONNECT_BY_PATH(Prenume || ' ' || Nume, ' -> ') AS Ierarhie
FROM
    Angajati
START WITH ManagerId IS NULL
CONNECT BY PRIOR ID_Angajat = ManagerId
ORDER BY Nivel_Ierarhic, ID_Sucursala, ID_Angajat;

```

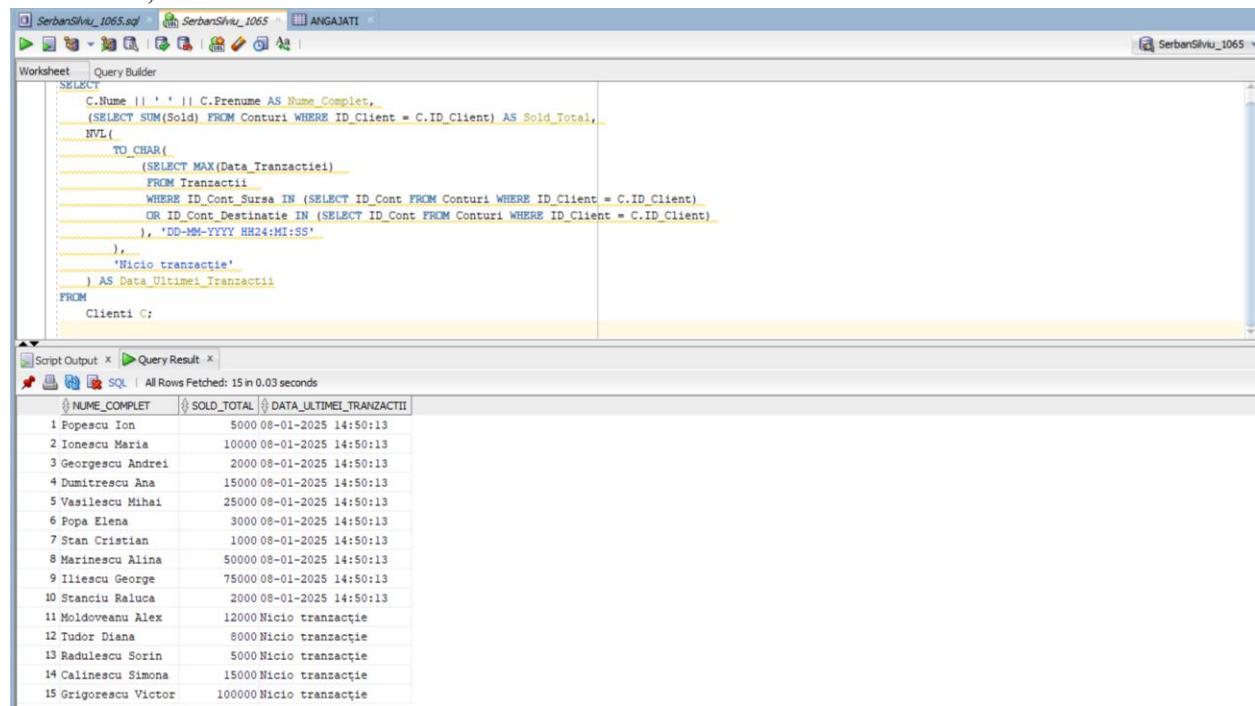
The screenshot shows the Oracle SQL Developer interface with the following details:

- Query Editor:** Contains the hierarchical query code.
- Script Output:** Shows the execution time: "All Rows Fetched: 31 in 0.081 seconds".
- Query Result:** Displays the hierarchical structure of employees. The columns are:

NIVEL_IERARHIC	ID_ANGAJAT	NUME_COMPLET	POZITIE	ID_SUCURSALA	MANAGERID	IERARHIE
1	1	O Silviu Serban	Director	1	(null)	-> Silviu Serban
2	2	1 Ion Popescu	Manager	1	0	-> Silviu Serban -> Ion Popescu
3	2	4 Ana Dumitrescu	Manager	2	0	-> Silviu Serban -> Ana Dumitrescu
4	2	7 Cristian Stan	Manager	3	0	-> Silviu Serban -> Cristian Stan
5	2	10 Raluca Stanciu	Manager	4	0	-> Silviu Serban -> Raluca Stanciu
6	2	13 Sorin Radulescu	Manager	5	0	-> Silviu Serban -> Sorin Radulescu
7	2	16 Adrian Popa	Manager	6	0	-> Silviu Serban -> Adrian Popa
8	2	19 Oana Mihai	Manager	7	0	-> Silviu Serban -> Oana Mihai
9	2	22 Marius Popescu	Manager	8	0	-> Silviu Serban -> Marius Popescu
10	2	25 Dan Iordache	Manager	9	0	-> Silviu Serban -> Dan Iordache
11	2	28 Alexandru Serban	Manager	10	0	-> Silviu Serban -> Alexandru Serban
12	3	2 Maria Ionescu	Casier	1	1	-> Silviu Serban -> Ion Popescu -> Maria Ionescu
13	3	3 Andrei Georgescu	Consultant Financiar	1	1	-> Silviu Serban -> Ion Popescu -> Andrei Georgescu
14	3	5 Mihai Vasilescu	Casier	2	4	-> Silviu Serban -> Ana Dumitrescu -> Mihai Vasilescu
15	3	6 Elena Popa	Consultant Financiar	2	4	-> Silviu Serban -> Ana Dumitrescu -> Elena Popa
16	3	8 Alina Marinescu	Casier	3	7	-> Silviu Serban -> Cristian Stan -> Alina Marinescu
17	3	9 George Iliescu	Consultant Financiar	3	7	-> Silviu Serban -> Cristian Stan -> George Iliescu
18	3	11 Alex Moldoveanu	Casier	4	10	-> Silviu Serban -> Raluca Stanciu -> Alex Moldoveanu
19	3	12 Diana Tudor	Consultant Financiar	4	10	-> Silviu Serban -> Raluca Stanciu -> Diana Tudor
20	3	14 Simona Calinescu	Casier	5	13	-> Silviu Serban -> Sorin Radulescu -> Simona Calinescu
21	3	15 Victor Grigorescu	Consultant Financiar	5	13	-> Silviu Serban -> Sorin Radulescu -> Victor Grigorescu
22	3	17 Gabriel Marin	Casier	6	16	-> Silviu Serban -> Adrian Popa -> Gabriel Marin
23	3	18 Roxana Enache	Consultant Financiar	6	16	-> Silviu Serban -> Adrian Popa -> Roxana Enache

14) Să se afișeze numele, prenumele, soldul total și data ultimei tranzacții pentru fiecare client.

```
SELECT
    C.Nume || ' ' || C.Prenume AS Nume_Complet,
    (SELECT SUM(Sold) FROM Conturi WHERE ID_Client = C.ID_Client) AS Sold_Total,
    NVL(
        TO_CHAR(
            (SELECT MAX(Data_Tranzactie)
             FROM Tranzactii
            WHERE ID_Cont_Sursa IN (SELECT ID_Cont FROM Conturi WHERE ID_Client =
C.ID_Client)
            OR ID_Cont_Destinatie IN (SELECT ID_Cont FROM Conturi WHERE ID_Client =
C.ID_Client)
            ), 'DD-MM-YYYY HH24:MI:SS'
        ),
        'Nicio tranzacție'
    ) AS Data_Ultimei_Tranzactii
FROM
    Clienti C;
```



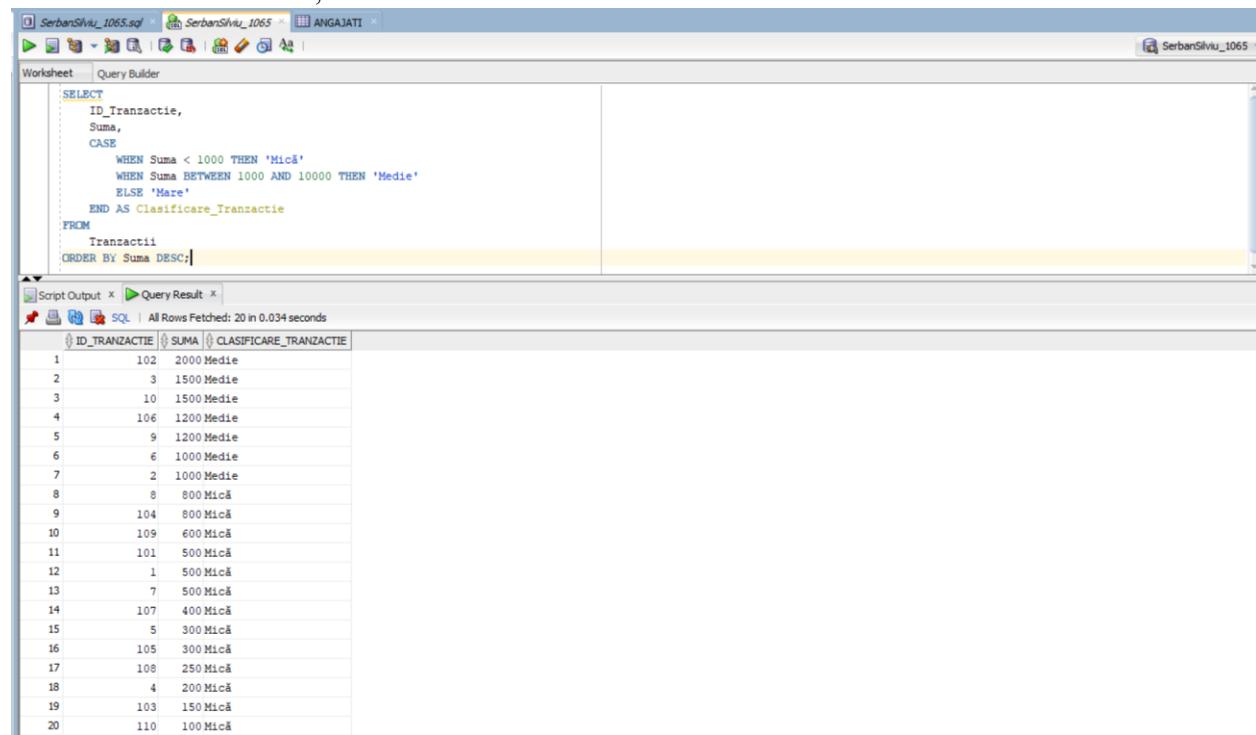
The screenshot shows the Oracle SQL Developer interface with the following details:

- Worksheet Tab:** Contains the SQL query from above.
- Script Output Tab:** Shows the execution status: "All Rows Fetched: 15 in 0.03 seconds".
- Query Result Tab:** Displays the results in a grid format with three columns: Nume_Complet, Sold_Total, and Data_Ultimei_Tranzactii.
- Data:**

Nume_Complet	Sold_Total	Data_Ultimei_Tranzactii
1 Popescu Ion	5000	08-01-2025 14:50:13
2 Ionescu Maria	10000	08-01-2025 14:50:13
3 Georgescu Andrei	2000	08-01-2025 14:50:13
4 Dumitrescu Ana	15000	08-01-2025 14:50:13
5 Vasilescu Mihai	25000	08-01-2025 14:50:13
6 Popa Elena	3000	08-01-2025 14:50:13
7 Stan Cristian	1000	08-01-2025 14:50:13
8 Marinescu Alina	50000	08-01-2025 14:50:13
9 Iliescu George	75000	08-01-2025 14:50:13
10 Stanciu Raluca	2000	08-01-2025 14:50:13
11 Moldoveanu Alex	12000	Nicio tranzactie
12 Tudor Diana	8000	Nicio tranzactie
13 Radulescu Sorin	5000	Nicio tranzactie
14 Calinescu Simona	15000	Nicio tranzactie
15 Grigorescu Victor	100000	Nicio tranzactie

15) Să se clasifice tranzacțiile în funcție de suma transferată :

```
SELECT
    ID_Tranzactie,
    Suma,
    CASE
        WHEN Suma < 1000 THEN 'Mică'
        WHEN Suma BETWEEN 1000 AND 10000 THEN 'Medie'
        ELSE 'Mare'
    END AS Clasificare_Tranzactie
FROM
    Tranzactii
ORDER BY Suma DESC;
```



The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for 'SerbanSilviu_1055.sql', 'SerbanSilviu_1065', and 'ANGAJATI'. Below the navigation bar is a toolbar with various icons. The main area is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the SQL query. The 'Query Result' pane displays the execution results in a table.

ID_TRANZACTIE	SUMA	CLASIFICARE_TRANZACTIE
1	102	2000 Medie
2	3	1500 Medie
3	10	1500 Medie
4	106	1200 Medie
5	9	1200 Medie
6	6	1000 Medie
7	2	1000 Medie
8	8	800 Mică
9	104	800 Mică
10	109	600 Mică
11	101	500 Mică
12	1	500 Mică
13	7	500 Mică
14	107	400 Mică
15	5	300 Mică
16	105	300 Mică
17	108	250 Mică
18	4	200 Mică
19	103	150 Mică
20	110	100 Mică

16) Să se creeze o vedere care să afișeze numele complet al clientilor, soldul total din conturi și sucursala unde au deschis contul.

```
CREATE OR REPLACE VIEW Vedere_Clienți_Solduri AS
```

```
SELECT
```

```
C.ID_Client,
C.Nume || ' ' || C.Prenume AS Nume_Complet,
(SELECT SUM(Sold) FROM Conturi WHERE ID_Client = C.ID_Client) AS Sold_Total,
S.Nume_Sucursala
FROM
Clienti C
JOIN Conturi Co ON C.ID_Client = Co.ID_Client
JOIN Sucursala S ON Co.ID_Sucursala = S.ID_Sucursala;
SELECT * FROM Vedere_Clienți_Solduri WHERE Sold_Total > 10000;
```

ID_CLIENT	NUME_COMPLET	SOLD_TOTAL	NUME_SUCURSALA
4	Dumitrescu Ana	15000	Iasi
5	Vasilescu Mihai	25000	Constanta
8	Marinescu Alina	50000	Craiova
9	Iliescu George	75000	Arad
11	Moldoveanu Alex	12000	Cluj
14	Calinescu Simona	15000	Constanta
15	Grigorescu Victor	100000	Brasov

17) Să se creeze o secvență pentru a genera automat ID-uri pentru tabela Angajati.

```
CREATE SEQUENCE SEQ_Angajati_ID START WITH 31 INCREMENT BY 1;
```

```
INSERT INTO Angajati (ID_Angajat, Prenume, Nume, CNP, ID_Sucursala, Pozitie, Data_Angajarii, Salariu, Telefon, Email)
```

```
VALUES ( SEQ_Angajati_ID.NEXTVAL, 'Maria', 'Popescu', '2920201123956', 1, 'Casier', SYSDATE, 3000, '0721234567', 'maria.popescu@gmail.com');
```

18) Să se identifice sucursala unde s-au deschis cele mai multe conturi în ultimul an și să se majoreze salariile angajaților din acea sucursală cu 5%.

UPDATE Angajati

SET Salariu = Salariu * 1.05

WHERE ID_Sucursala = (

```
    SELECT c.ID_Sucursala
    FROM Conturi c
    WHERE c.Data_Deschiderii >= ADD_MONTHS(SYSDATE, -12)
    GROUP BY c.ID_Sucursala
    ORDER BY COUNT(*) DESC
    FETCH FIRST 1 ROWS ONLY
);
```

```
Script Output x Query Result x
Task completed in 0.061 seconds
4 rows updated.

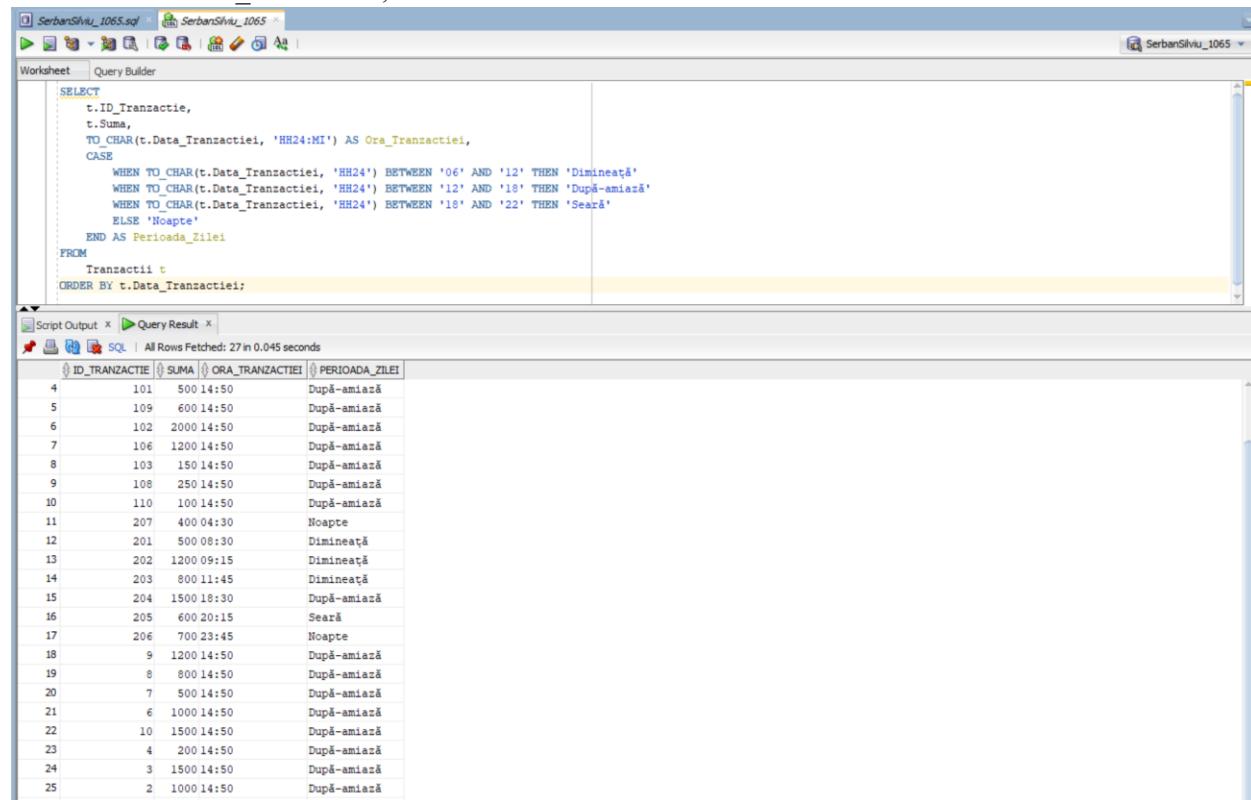
SELECT ID_Angajat, Nume, Prenume, ID_Sucursala, Salariu
FROM Angajati
WHERE ID_Sucursala = (
    SELECT c.ID_Sucursala
    FROM Conturi c
    WHERE c.Data_Deschiderii >= ADD_MONTHS(SYSDATE, -12)
    GROUP BY c.ID_Sucursala
    ORDER BY COUNT(*) DESC
    FETCH FIRST 1 ROWS ONLY
);
```

ID_ANGAJAT	NUME	PRENUME	ID_SUCURSALA	SALARIU	
1	0	Serban	Silviu	1	10500
2	1	Popescu	Ion	1	5250
3	2	Ionescu	Maria	1	3150
4	3	Georgescu	Andrei	1	3675

19) Să se afișeze tranzacțiile clasificate în funcție de perioada zilei: Dimineată: 06:00 - 12:00
 După-amiază: 12:01 - 18:00 Seară: 18:01 - 22:00 Noapte: 22:01 - 05:59 .

```

SELECT
    t.ID_Tranzactie,
    t.Suma,
    TO_CHAR(t.Data_Tranzactiei, 'HH24:MI') AS Ora_Tranzactiei,
    CASE
        WHEN TO_CHAR(t.Data_Tranzactiei, 'HH24') BETWEEN '06' AND '12' THEN 'Dimineată'
        WHEN TO_CHAR(t.Data_Tranzactiei, 'HH24') BETWEEN '12' AND '18' THEN 'După-amiază'
        WHEN TO_CHAR(t.Data_Tranzactiei, 'HH24') BETWEEN '18' AND '22' THEN 'Seară'
        ELSE 'Noapte'
    END AS Perioada_Zilei
FROM
    Tranzactii t
ORDER BY t.Data_Tranzactiei;
    
```



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

- Query Editor:** Contains the SQL query provided above.
- Script Output:** Shows the execution message: "All Rows Fetched: 27 in 0.045 seconds".
- Query Result:** Displays a table with 27 rows, showing the results of the query. The columns are labeled: ID_TRANZACTIE, SUMA, ORA_TRANZACTIEI, and PERIOADA_ZILEI.

ID_TRANZACTIE	SUMA	ORA_TRANZACTIEI	PERIOADA_ZILEI
4	101	500 14:50	După-amiază
5	109	600 14:50	După-amiază
6	102	2000 14:50	După-amiază
7	106	1200 14:50	După-amiază
8	103	150 14:50	După-amiază
9	108	250 14:50	După-amiază
10	110	100 14:50	După-amiază
11	207	400 04:30	Noapte
12	201	500 08:30	Dimineată
13	202	1200 09:15	Dimineată
14	203	800 11:45	Dimineată
15	204	1500 18:30	După-amiază
16	205	600 20:15	Seară
17	206	700 23:45	Noapte
18	9	1200 14:50	După-amiază
19	8	800 14:50	După-amiază
20	7	500 14:50	După-amiază
21	6	1000 14:50	După-amiază
22	10	1500 14:50	După-amiază
23	4	200 14:50	După-amiază
24	3	1500 14:50	După-amiază
25	2	1000 14:50	După-amiază

20) Să se afișeze clienții care au realizat tranzacții de tip Transfer, Depunere și Retragere în ultimele 3 luni. Pentru fiecare client, afișează numele complet și numărul total de tranzacții pentru fiecare tip.

SELECT

```

    cl.ID_Client,
    cl.Nume || ' ' || cl.Prenume AS Nume_Complet,
    COUNT(CASE WHEN t.Tip_Tranzactie = 'Transfer' THEN 1 END) AS Numar_Transferuri,
    COUNT(CASE WHEN t.Tip_Tranzactie = 'Depunere' THEN 1 END) AS Numar_Depunerri,
    COUNT(CASE WHEN t.Tip_Tranzactie = 'Retragere' THEN 1 END) AS Numar_Retrageri
FROM
    Clienti cl
JOIN Conturi c ON cl.ID_Client = c.ID_Client
LEFT JOIN Tranzactii t ON c.ID_Cont = t.ID_Cont_Sursa
WHERE
    t.Data_Tranzactiei >= ADD_MONTHS(SYSDATE, -3)
GROUP BY
    cl.ID_Client, cl.Nume, cl.Prenume
HAVING
    COUNT(CASE WHEN t.Tip_Tranzactie = 'Transfer' THEN 1 END) > 0
    AND COUNT(CASE WHEN t.Tip_Tranzactie = 'Depunere' THEN 1 END) > 0
    AND COUNT(CASE WHEN t.Tip_Tranzactie = 'Retragere' THEN 1 END) > 0;

```

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

- Worksheet Tab:** Contains the SQL query from above.
- Script Output Tab:** Shows the execution message: "All Rows Fetched: 1 in 0.066 seconds".
- Query Result Tab:** Displays the result of the query in a table format.

ID_CLIENT	NUME_COMPLET	NUMAR_TRANSFERURI	NUMAR_DEPUNERI	NUMAR_RETRAGERI
1	3 Georgescu Andrei	2	2	1