

1. Création de la base de données

```
DROP TABLE Operation;
DROP TABLE Clients;
DROP TABLE Compte;
```

```
CREATE TABLE Compte
(
    NumCompte      NUMBER(6) PRIMARY KEY,
    Solde_initial  NUMBER(8,2) NOT NULL,
    Total_Debit    NUMBER(8,2) DEFAULT 0,
    Total_Credit   NUMBER(8,2) DEFAULT 0,
    Solde_Actuelle NUMBER(8,2) DEFAULT 0
);
```

```
CREATE TABLE Clients
(
    NumClient  NUMBER(5) PRIMARY KEY,
    Nom        VARCHAR2(20) NOT NULL,
    Prenom     VARCHAR2(20) NOT NULL,
    Adresse    VARCHAR2(20),
    Telephone  NUMBER(10) NOT NULL,
    email      VARCHAR2(30) NOT NULL,
    NumCompte  NUMBER(6),
    CONSTRAINT fk_compte_client FOREIGN KEY (NumCompte)
    REFERENCES Compte(NumCompte)
);
```

```
CREATE TABLE Operation
(
    NumOper      NUMBER(5) PRIMARY KEY,
    Ncompte      NUMBER(6),
    Date_Oper    DATE NOT NULL,
    Debit        NUMBER(8,2) NOT NULL,
    Credit       NUMBER(8,2) NOT NULL,
    CONSTRAINT fk_compte_operation FOREIGN KEY (Ncompte)
    REFERENCES Compte (NumCompte)
);
```

2. Insertion des clients, des comptes et des opérations

```
INSERT INTO Compte VALUES (407789,1000,0,0,0);
INSERT INTO Compte VALUES (401065,2500,0,0,0);

INSERT INTO Clients VALUES
(1, 'DIB', 'DJAMEL', 'SETIF', 05698659, 'dj@yahoo.fr', 407789);
INSERT INTO Clients VALUES
(2, 'SALEM', 'MALIKA', 'SETIF', 05698759, 'sm@yahoo.fr', 401065);

INSERT INTO Operation VALUES (1,407789, '02/10/2017', 100,0);
INSERT INTO Operation VALUES (2,401065, '05/10/2017', 0,250);
INSERT INTO Operation VALUES (3,407789, '10/10/2017', 0,500);
INSERT INTO Operation VALUES (4,407789, '13/10/2017', 300,0);
INSERT INTO Operation VALUES (5,401065, '13/10/2017', 0,600);
COMMIT;
```

3. Affichage des tables

```
SELECT * FROM Compte;  
SELECT * FROM Clients;  
SELECT * FROM Operation;
```

Mise à jour des soldes actuels des comptes 407789 et 401065

$$\text{Solde_Actuelle} = \text{Solde_initial} + \sum_{i=1}^n \text{debit}_i + \sum_{i=1}^n \text{credit}_i$$

DECLARE

```
v_total_debit Operation.Debit%TYPE;  
v_total_credit Operation.Credit%TYPE;  
v_solde_actuelle Compte.Solde_Actuelle%TYPE;  
v_solde_init Compte.Solde_Initial%TYPE;
```

• CLIENT 1

```
SELECT SUM(Debit) INTO v_total_debit  
FROM Operation WHERE Ncompte = 407789;
```

```
SELECT SUM(Credit) INTO v_total_credit  
FROM Operation WHERE Ncompte = 407789;
```

```
SELECT Solde_Initial INTO v_solde_init  
FROM Compte WHERE NumCompte=407789;
```

```
v_solde_actuelle: = v_solde_init - v_total_debit + v_total_credit;
```

```
UPDATE Compte  
SET Solde_Actuelle = v_solde_actuelle  
WHERE NumCompte = 407789;
```

• CLIENT 2

```
SELECT SUM(Debit) INTO v_total_debit  
FROM Operation WHERE Ncompte = 401065;
```

```
SELECT SUM(Credit) INTO v_total_credit  
FROM Operation WHERE Ncompte = 401065;
```

```
SELECT Solde_Initial INTO solde_init  
FROM Compte WHERE NumCompte=401065;
```

```
v_solde_actuelle: = v_solde_init - v_total_debit + v_total_credit;
```

```
UPDATE Compte  
SET Solde_Actuelle = v_solde_actuelle  
WHERE NumCompte = 401065 ;
```

4. Procédure de transfert comptel vers compte2 :

```
CREATE OR REPLACE PROCEDURE Transf_Comptel_Compte2 (
    comptel IN Compte.NumCompte%TYPE,
    compte2 IN Compte.NumCompte%TYPE,
    montant IN Compte.Solde_initial%TYPE)
IS
    v_num_comptel Compte.NumCompte%TYPE;
    v_num_compte2 Compte.NumCompte%TYPE;
    v_solde_actuelle_debiteur Compte.Solde_actuelle%TYPE;
    v_solde_actuelle_crediteur Compte.Solde_actuelle%TYPE ;
    v_numopr Operation.NumOper%TYPE;
    v_date_operation Operation.Date_Oper%TYPE;
    Compte_crediteur_inexistant EXCEPTION ;
    Compte_debiteur_inexistant EXCEPTION ;
    Solde_insuffisant EXCEPTION ;

BEGIN
    -- Vérification de compte 1
    SELECT COUNT (*) INTO v_num_comptel
    FROM Clients WHERE NumCompte = comptel;
    IF v_num_comptel=0
        RAISE Compte_crediteur_inexistant ;
    END IF;
    ELSE
        -- Vérification de compte 2
        SELECT COUNT (*) INTO v_num_compte2
        FROM Clients WHERE NumCompte = compte2 ;
        IF v_num_compte2=0
            RAISE Compte_debiteur_inexistant ;
        END IF;
        ELSE
            -- solde actuelle de compte 1.
            SELECT Solde_actuelle INTO v_solde_actuelle_debiteur
            FROM Compte WHERE NumCompte = comptel ;
            -- Récupération du solde actuelle du créditeur.
            SELECT Solde_actuelle INTO v_solde_actuelle_crediteur
            FROM Compte WHERE NumCompte = compte2 ;
            -- Si le solde actuel de créditeur est suffisant
            IF v_solde_actuelle_debiteur >= montant THEN
                -- nouveau numéro d'opération = dernier numéro d'opération +1.
                SELECT MAX(NumOper) INTO v_numopr FROM Operation;
                v_numopr = v_numopr +1;
                -- mise à jour des comptes débiteur et créditeur
                UPDATE Compte SET Solde_actuelle = Solde_actuelle - montant
                WHERE NumCompte = comptel ;
                v_solde_actuelle_debiteur = v_solde_actuelle_debiteur - montant ;

                UPDATE Compte SET Solde_actuelle = Solde_actuelle + montant
                WHERE NumCompte = compte2 ;
                v_solde_actuelle_crediteur=v_solde_actuelle_crediteur + montant ;
                -- Insertion de deux opérations créditrices et débiteur
                v_date_operation: = TO_DATE(CURRENT_DATE, 'dd/mm/yyyy');

                INSERT INTO Operation VALUES
                    (v_numopr, comptel, v_date_operation, montant, 0);
            END IF;
        END IF;
    END IF;
END;
```

```

v_numopr = v_numopr +1;

INSERT INTO Operation VALUES
(v_numopr, compte2, v_date_operation, 0, montant);

-- Affichage des nouveaux soldes
dbms_output.put_line('Solde Actuelle du compte 1 : '||
v_solde_actuelle_debiteur);

dbms_output.put_line ('Solde Actuelle du compte 2 : '||
v_solde_actuelle_crediteur) ;

ELSE
RAISE Solde_insuffisant;
END IF;
EXCEPTION
WHEN Solde_insuffisance THEN
RAISE_APPLICATION_ERROR (-20000,'Solde Insuffisant') ;

WHEN Compte_crediteur_inexistant THEN
RAISE_APPLICATION_ERROR (-20000,'Compte créditeur inexistant') ;

WHEN Compte_debiteur_inexistant THEN
RAISE_APPLICATION_ERROR (-20000, 'Compte débiteur inexistant') ;

END TRANSF_COMPTE1_COMPTE2 ;

```

5. Déclencheur de mise à jour des totaux des comptes créditeur débiteur :

```

CREATE OR REPLACE TRIGGER Declencheur_TotalCreditDebit
BEFORE INSERT ON Operation
FOR EACH ROW
DECLARE
v_total_credit Compte.Total_credit%TYPE;
v_total_debit Compte.Total_Debit%TYPE;
BEGIN
SELECT SUM(credit) INTO v_total_credit
FROM Operation
WHERE Ncompte = :new.Ncompte;

SELECT SUM(debit) INTO v_total_debit
FROM Operation
WHERE Ncompte = :new.Ncompte;

UPDATE Compte
SET Total_Credit= v_total_credit +:new.credit,
WHERE NumCompte = :new.Ncompte;

UPDATE Compte
SET Total_Debit= v_total_debit +:new.debit,
WHERE NumCompte = :new.Ncompte;

END Declencheur_TotalCreditDebit;

```