controle1.sql

```sql
CREATE DATABASE banque
    DEFAULT CHARACTER SET = 'utf8mb4';

use banque;
--A--1
CREATE TABLE compte(
    NCompte INT PRIMARY KEY,
    solde REAL,
    seuil REAL
);
CREATE TABLE mouvement(
    NMouvement INT AUTO_INCREMENT PRIMARY KEY,
    NCompte INT,
    montant REAL,
    sens CHAR(1),
    date_mvt DATETIME,
    FOREIGN KEY(NCompte) REFERENCES compte(NCompte),
    CONSTRAINT CK_Sens CHECK (sens IN ('+', '-'))
);
--B--2
SELECT NCompte,SUM(montant) from mouvement
WHERE sens='+' and YEAR(date_mvt)=YEAR(CURRENT_DATE)
GROUP BY NCompte;

--B--3
SELECT MAX(date_mvt) from mouvement
GROUP BY NCompte;

--C--4
DELIMITER//
CREATE FUNCTION total_mouvement(date_debut DATE,date_fin DATE,NCompte INT)
RETURNS REAL
BEGIN
READS SQL DATA
BEGIN
DECLARE total_mvt REAL
DECLARE total_credit REAL
DECLARE total_debit REAL
SET total_credit=(SELECT SUM(monatant) from mouvement WHERE sens='+' AND NCompte=NCompte AND date_mvt BETWEEN date_debut AND date_fin)
SET total_debit=(SELECT SUM(monatant) from mouvement WHERE sens='-' AND NCompte=NCompte AND date_mvt BETWEEN date_debut AND date_fin)
SET total_mvt=total_credit-total_debit
RETURN total_mvt;
END
DELIMITER;
--D--5
DELIMITER//
CREATE PROCEDURE ps_crediter(IN num_compte INT,IN montant REAL,OUT nouveau_solde REAL)
BEGIN
    DECLARE compte_existe INT;
    SELECT COUNT(*) INTO compte_existe
    FROM Compte
    WHERE NCompte = num_compte;

    IF compte_existe = 0 THEN
        SET nouveau_solde = -1;
    ELSE
            BEGIN
            DECLARE continue_handler INT DEFAULT 0;
            DECLARE EXIT HANDLER FOR SQLEXCEPTION
            SET continue_handler = 1;

            START TRANSACTION;

            INSERT INTO Mouvement (NCompte, montant, sens, date_mvt)
            VALUES (num_compte, montant, '+', CURDATE());

            IF continue_handler = 0 THEN
                UPDATE Compte
                SET solde = solde + montant,
                    date_dernier_mouvement = CURDATE()
                WHERE NCompte = num_compte;
                COMMIT;
                SELECT solde INTO nouveau_solde FROM Compte WHERE NCompte = num_compte;
            ELSE
                ROLLBACK;
                SET nouveau_solde = -2;
            END IF;
        END;
    END IF;
END //

DELIMITER ;
--D--6
DELIMITER //

CREATE PROCEDURE ps_debiter(
    IN num_compte INT,
    IN montant DECIMAL(10, 2),
    OUT nouveau_solde DECIMAL(10, 2)
)
BEGIN
    DECLARE compte_exist INT;
    DECLARE solde DECIMAL(10, 2);

    SELECT COUNT(*) INTO compte_exist
    FROM Compte
    WHERE NCompte = num_compte;

    IF compte_exist = 0 THEN
        SET nouveau_solde = -1;
    ELSE
        SELECT solde INTO solde
        FROM Compte
        WHERE NCompte = num_compte;

        IF montant > solde THEN
            SET nouveau_solde = -2;
        ELSE
            BEGIN
                DECLARE continue_handler INT DEFAULT 0;
                DECLARE EXIT HANDLER FOR SQLEXCEPTION
                SET continue_handler = 1;

                START TRANSACTION;

                INSERT INTO Mouvement (NCompte, montant, sens, date_mouvement)
                VALUES (num_compte, montant, '-', CURDATE());

                IF continue_handler = 0 THEN
                    UPDATE Compte
                    SET solde = solde - montant,
                        date_dernier_mouvement = CURDATE()
                    WHERE NCompte = num_compte;
                    COMMIT;
                    SELECT solde INTO nouveau_solde FROM Compte WHERE NCompte = num_compte;
                ELSE
                    ROLLBACK;
                    SET nouveau_solde = -3;
                END IF;
            END;
        END IF;
    END IF;
END //

DELIMITER ;

INSERT INTO compte (NCompte, solde)
VALUES (1, 1000),
(2, 2000),
(3, 3000),
(4, 4000),
(5, 5000),
(6, 6000);

INSERT INTO mouvement (NCompte, montant, sens, date_mvt)
VALUES (1, 50, '+', '2024-01-19'),
(1, 50, '+', '2024-02-19'),
(1, 60, '+', '2024-03-19'),
(1, 100, '+', '2022-02-19'),
(1, 200, '+', '2024-01-19'),
(1, 50, '+', '2024-01-19');
```