

contrôle1.sql

```
1 CREATE DATABASE banque
2     DEFAULT CHARACTER SET = 'utf8mb4';
3
4 use banque;
5 --A--1
6 CREATE TABLE compte(
7     NCompte INT PRIMARY KEY,
8     solde REAL,
9     seuil REAL
10 );
11 CREATE TABLE mouvement(
12     NMouvement INT AUTO_INCREMENT PRIMARY KEY,
13     NCompte INT,
14     montant REAL,
15     sens CHAR(1),
16     date_mvt DATETIME,
17     FOREIGN KEY(NCompte) REFERENCES compte(NCompte),
18     CONSTRAINT CK_Sens CHECK (sens IN ('+', '-'))
19 );
20 --B--2
21 SELECT NCompte,SUM(montant) from mouvement
22 WHERE sens='+' and YEAR(date_mvt)=YEAR(CURRENT_DATE)
23 GROUP BY NCompte;
24
25 --B--3
26 SELECT MAX(date_mvt) from mouvement
27 GROUP BY NCompte;
28
29 --C--4
30 DELIMITER//
31 CREATE FUNCTION total_mouvement(date_debut DATE,date_fin DATE,NCompte INT)
32 RETURNS REAL
33 BEGIN
34     READS SQL DATA
35     BEGIN
36         DECLARE total_mvt REAL
37         DECLARE total_credit REAL
38         DECLARE total_debit REAL
39         SET total_credit=(SELECT SUM(monatant) from mouvement WHERE sens='+' AND NCompte=NCompte
40 AND date_mvt BETWEEN date_debut AND date_fin)
41         SET total_debit=(SELECT SUM(monatant) from mouvement WHERE sens='-' AND NCompte=NCompte
42 AND date_mvt BETWEEN date_debut AND date_fin)
43         SET total_mvt=total_credit-total_debit
44         RETURN total_mvt;
45     END
46 DELIMITER;
47 --D--5
48 DELIMITER//
49 CREATE PROCEDURE ps_crediter(IN num_compte INT,IN montant REAL,OUT nouveau_solde REAL)
50 BEGIN
51     DECLARE compte_existe INT;
52     SELECT COUNT(*) INTO compte_existe
```

```
51     FROM Compte
52     WHERE NCompte = num_compte;
53
54     IF compte_existe = 0 THEN
55         SET nouveau_solde = -1;
56     ELSE
57         BEGIN
58             DECLARE continue_handler INT DEFAULT 0;
59             DECLARE EXIT HANDLER FOR SQLEXCEPTION
60                 SET continue_handler = 1;
61
62             START TRANSACTION;
63
64             INSERT INTO Mouvement (NCompte, montant, sens, date_mvt)
65             VALUES (num_compte, montant, '+', CURDATE());
66
67             IF continue_handler = 0 THEN
68                 UPDATE Compte
69                 SET solde = solde + montant,
70                     date_dernier_mouvement = CURDATE()
71                 WHERE NCompte = num_compte;
72                 COMMIT;
73                 SELECT solde INTO nouveau_solde FROM Compte WHERE NCompte = num_compte;
74             ELSE
75                 ROLLBACK;
76                 SET nouveau_solde = -2;
77             END IF;
78         END;
79     END IF;
80 END //
```

```
81
82 DELIMITER ;
83 --D--6
84 DELIMITER //
```

```
85
86 CREATE PROCEDURE ps_debiter(
87     IN num_compte INT,
88     IN montant DECIMAL(10, 2),
89     OUT nouveau_solde DECIMAL(10, 2)
90 )
91 BEGIN
92     DECLARE compte_exist INT;
93     DECLARE solde DECIMAL(10, 2);
94
95     SELECT COUNT(*) INTO compte_exist
96     FROM Compte
97     WHERE NCompte = num_compte;
98
99     IF compte_exist = 0 THEN
100         SET nouveau_solde = -1;
101     ELSE
102         SELECT solde INTO solde
103         FROM Compte
104         WHERE NCompte = num_compte;
105
```

```
106         IF montant > solde THEN
107             SET nouveau_solde = -2;
108         ELSE
109             BEGIN
110                 DECLARE continue_handler INT DEFAULT 0;
111                 DECLARE EXIT HANDLER FOR SQLEXCEPTION
112                     SET continue_handler = 1;
113
114                 START TRANSACTION;
115
116                 INSERT INTO Mouvement (NCompte, montant, sens, date_mouvement)
117                     VALUES (num_compte, montant, '-', CURDATE());
118
119                 IF continue_handler = 0 THEN
120                     UPDATE Compte
121                         SET solde = solde - montant,
122                             date_dernier_mouvement = CURDATE()
123                         WHERE NCompte = num_compte;
124                     COMMIT;
125                     SELECT solde INTO nouveau_solde FROM Compte WHERE NCompte =
num_compte;
126                 ELSE
127                     ROLLBACK;
128                     SET nouveau_solde = -3;
129                 END IF;
130             END;
131         END IF;
132     END IF;
133 END //
```

```
134
135 DELIMITER ;
136
137 INSERT INTO compte (NCompte, solde)
138 VALUES (1, 1000),
139 (2, 2000),
140 (3, 3000),
141 (4, 4000),
142 (5, 5000),
143 (6, 6000);
144
145 INSERT INTO mouvement (NCompte, montant, sens, date_mvt)
146 VALUES (1, 50, '+', '2024-01-19'),
147 (1, 50, '+', '2024-02-19'),
148 (1, 60, '+', '2024-03-19'),
149 (1, 100, '+', '2022-02-19'),
150 (1, 200, '+', '2024-01-19'),
151 (1, 50, '+', '2024-01-19');
```