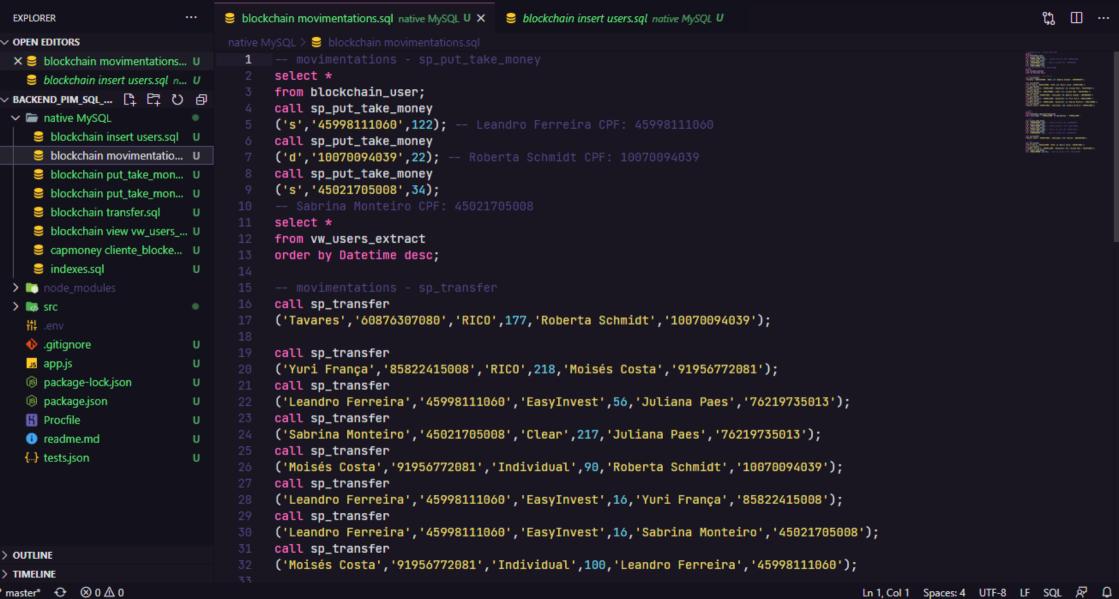
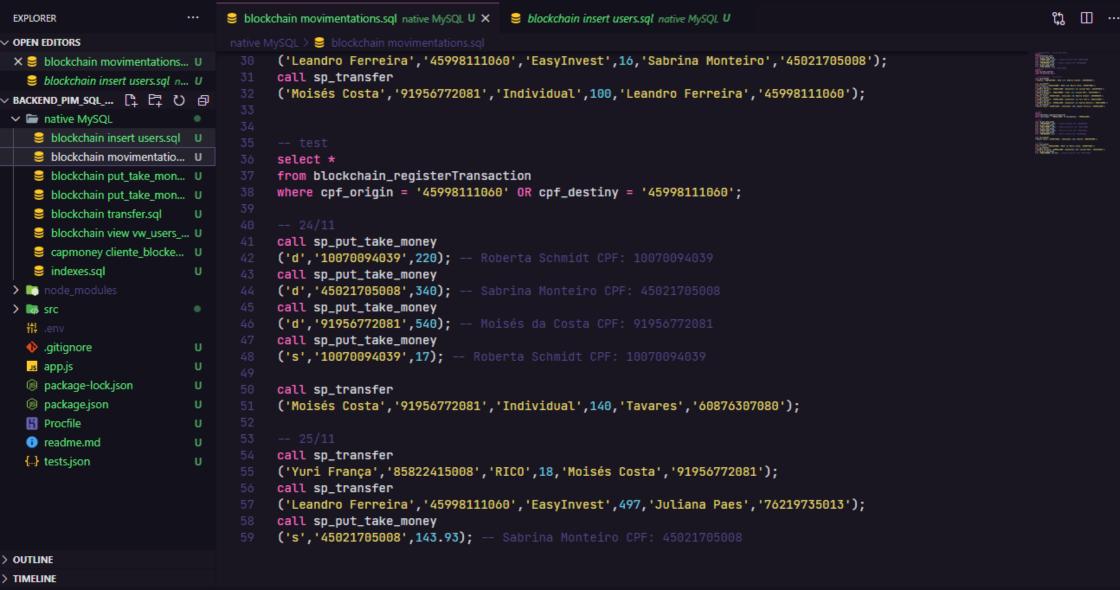
```
native MySQL > 

blockchain transfer.sql
       desc blockchain_registerTransaction;
       START TRANSACTION;
       DROP PROCEDURE IF EXISTS sp_transfer;
       DELIMITER $$
       CREATE PROCEDURE sp_transfer (
                IN fromClient VARCHAR(50)
                , IN fromCPF
                                VARCHAR(15)
               ,IN fromAgency VARCHAR(30)
               ,IN p_value
                                DECIMAL(8,2)
               ,IN toClient
                                VARCHAR (30)
                                VARCHAR(15)
               ,IN TOcpf
       BEGIN
           IF p_value > (select balance from blockchain_user where cpf = fromCPF) THEN
               SELECT 'OPS! Valor intencionado é maior do que o valor do saldo atual na conta!' as 'STOP!';
           ELSE
                        UPDATE blockchain_user
                            SET balance = balance - p_value
                        WHERE cpf = fromCPF;
                        UPDATE blockchain_user
                            SET balance = balance + p_value
                        WHERE cpf = ToCPF;
               INSERT INTO blockchain_registerTransaction VALUES
                    (default fromClient fromCPF fromAgency n value toClient toCPF now() now()).
          ⊗ 0 \( \Delta \) 0
                                                                                                                          Ln 1. Col 1 Tab Size: 4 UTF-8 LF SOL ₽ 🚨
master*
      ↔
```

```
ELSE
               UPDATE blockchain_user
                    SET balance = balance - p_value
                WHERE cpf = fromCPF;
               UPDATE blockchain_user
                    SET balance = balance + p_value
                WHERE cpf = ToCPF;
        INSERT INTO blockchain_registerTransaction VALUES
            (default, fromClient, fromCPF, fromAgency, p_value, toClient, toCPF, now(), now());
        SELECT CONCAT('Transferência entre Sr(a): ', fromClient , ' para CPF: ', toCPF , ' com valor de: R$', p_value , 'realizada co
    END IF:
END $$
DELIMITER ;
SELECT name, cpf, agency, balance from blockchain_user;
call sp_transfer('Tavares','60876307080','RICO',250,'Sabrina Monteiro','45021705008');
SELECT name, cpf, agency, balance from blockchain_user;
select * from blockchain_registerTransaction;
ROLLBACK;
COMMIT;
```

SELECT OLS: AUTOL THICHICTOHOUG & MOTOL OF AGE A AUTOL OF SULFOR UP CONTO: US STOL: "





```
→ BEGIN

         -- getting primarykey
20
         DECLARE v user int;
21
      SELECT id
      from blockchain user
      where cpf = p toCPF
24
25
      into v_user;
26
    27
      -- efetuar depósito (put)
28
      UPDATE blockchain user
                   SET balance = balance + p_value
30
                WHERE cpf = p toCPF;
31
      -- creating log
32
33
      INSERT INTO blockchain_extract
34
      values
         (default, 'depósito', p_value, v_user, now(), now() );
35
36
      SELECT CONCAT('Depósito para CPF:', p_toCPF, ' realizado com sucesso') as 'Success';
37
38
      ELSEIF p_action = UPPER
39
      ('s') THEN
      -- verificando se a quantidade é maior que o valor atual do saldo
41
    from blockchain user
```

```
□ □ □ | 7 7 9 0 | 20 | 20 | 20 | 10 | 1000 rows
▼ | ★ | ◆ ○ ○ □ □
       ELSEIF p action = UPPER
39
       ('s') THEN
40
        -- verificando se a quantidade é maior que o valor atual do saldo
41
       IF p value > (select balance
42
        from blockchain user
43
        WHERE cpf = p toCPF) THEN
44
       SELECT 'O valor solicitado para saque é maior que o valor do saldo atual,
45
                                   entre com um valor menor para efetuar o saque.' as 'WARNING';
46
       ELSE
47
        -- efetuar saque
49
       UPDATE blockchain user
                               SET balance = balance - p value
50
                           WHERE cpf = p toCPF;
51
        -- creating log
52
        INSERT INTO blockchain extract
53
       values
54
           (default, 'saque', p_value, v_user, now(), now() );
55
56
57
        SELECT CONCAT('Saque realizado com sucesso para o CPF: ', p toCPF );
       END
58
59
       IF;
            ELSE
61
62
                IF p action NOT IN ( UPPER('D'), UPPER('S') ) THEN
```

```
DELIMITER $
16 V CREATE TRIGGER trg_client_blocked
        AFTER DELETE ON capmonev_client
        FOR EACH ROW
            BEGIN
                INSERT INTO capmoney_clientBlocked
                (id_client,name, cpf, bornAt, email, phone, password,address,category,agency, account, openAt,createdAt,updatedAt)
                VALUES
                (old.id, old.name, old.cpf, old.bornAt, old.email, old.phone, old.password, old.address, old.category, old.agency,
    END $
    select * from capmoney_clientBlocked;
    select * from capmoney_client;
    SELECT * FROM capmoney_clientBlocked;
    ROLLBACK;
    COMMIT;
```

START TRANSACTION: