

2.5 - Apresente a procedure e o trigger criados, explicando o funcionamento de cada um.

Procedure

```
CREATE OR REPLACE PROCEDURE insert_transaction(
    p_account_id bigint,
    p_amount numeric(15,2),
    p_transaction_date timestamp with time zone
)
LANGUAGE plpgsql
AS $$
DECLARE
    v_old_balance numeric(15,2);
    v_new_balance numeric(15,2);
BEGIN
    BEGIN
        SELECT balance INTO v_old_balance
        FROM accounts
        WHERE account_id = p_account_id
        FOR UPDATE;

        v_new_balance := v_old_balance + p_amount;

        UPDATE accounts
        SET balance = v_new_balance
        WHERE account_id = p_account_id;

        INSERT INTO transactions (account_id, amount, transaction_date)
        VALUES (p_account_id, p_amount, p_transaction_date);

        INSERT INTO audit_log (transaction_id, account_id, old_balance,
            new_balance, change_date)
        VALUES (currval(pg_get_serial_sequence('transactions', 'transaction_id')),
            p_account_id, v_old_balance, v_new_balance, p_transaction_date);

        COMMIT;
    EXCEPTION
        WHEN OTHERS THEN
            ROLLBACK;
            RAISE;
    END;
END;
$$;
```

A **procedure** `insert_transaction` é responsável por inserir uma nova transação, atualizar o saldo da conta e registrar a alteração no log de auditoria, tudo dentro de uma transação para garantir a integridade dos dados. Ela utiliza **BEGIN**, **COMMIT**, e **ROLLBACK** para garantir que, em caso de erro, o banco de dados não tenha dados inconsistentes.

Trigger

```
CREATE OR REPLACE FUNCTION after_transaction_insert()
RETURNS TRIGGER AS $$
DECLARE
    v_old_balance numeric(15,2);
    v_new_balance numeric(15,2);
BEGIN
    SELECT balance INTO v_old_balance
    FROM accounts
    WHERE account_id = NEW.account_id
    FOR UPDATE;

    v_new_balance := v_old_balance + NEW.amount;

    UPDATE accounts
    SET balance = v_new_balance
    WHERE account_id = NEW.account_id;

    INSERT INTO audit_log (transaction_id, account_id, old_balance, new_balance,
change_date)
    VALUES (NEW.transaction_id, NEW.account_id, v_old_balance, v_new_balance,
NEW.transaction_date);

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_after_transaction_insert
AFTER INSERT ON transactions
FOR EACH ROW
EXECUTE FUNCTION after_transaction_insert();
```

O **trigger** `trigger_after_transaction_insert` é acionado após a inserção de uma transação na tabela `transactions`. Ele automaticamente atualiza o saldo da conta e insere o log de auditoria. Este trigger assegura que as mudanças necessárias no saldo e na auditoria sejam feitas sem a intervenção explícita do desenvolvedor na lógica de negócios.