# Exercício trigger 2

Execute a sql abaixo para criar a tabela

CREATE TABLE conta(
id INTEGER,
id\_cliente INTEGER,
saldo NUMERIC(15,2),
limite NUMERIC(15,2),
cheque\_especial NUMERIC(15,2),
PRIMARY KEY(id))

A seguir, execute os seguintes comandos

INSERT INTO conta values (1,1,1200.00,500.00,0.0)

Crie outra tabela, que armazenará saques realizados:

CREATE TABLE saque( id INTEGER, id\_conta INTEGER, valor NUMERIC(15,2), PRIMARY KEY(id))

Exercício: crie uma função trigger que será acionada ao inserir na tabela saque. Essa função deverá fazer o controle de verificação de saque da conta, isto é, se o valor(NEW) na tabela saque for menor ou igual ao saldo da tabela conta, permite o saque e atualiza a tabela conta.

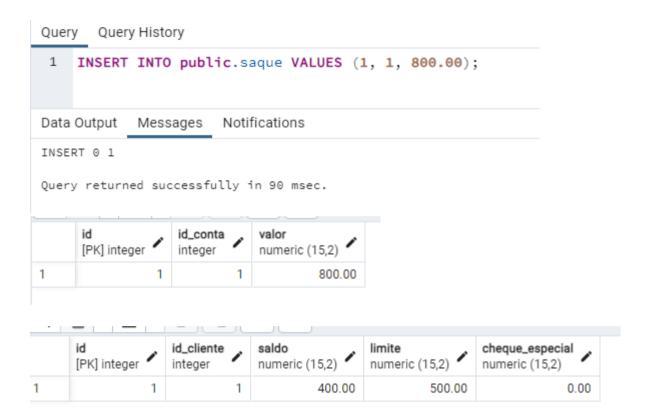
Se o valor(NEW) de saque for maior que o valor de saldo na tabela conta, então verifica-se se o valor(NEW) de saque é maior que o saldo+limite-cheque\_especial, se sim, não permite a inserção na tabela saque (cliente não possui limite disponível).

Se o valor(NEW) de saque for maior que o valor de saldo na tabela conta e o valor(NEW) do saque é menor ou igual que o saldo+limite-cheque\_especial, se sim, atualiza-se o saldo para 0 e o excedente deve ser colocado no campo cheque\_especial (cliente possui limite disponível).

```
CREATE OR REPLACE FUNCTION verifica_saque() RETURNS TRIGGER AS
$verifica saque$
DECLARE
  saldo atual NUMERIC(15, 2);
  disponivel NUMERIC(15, 2);
BEGIN
  SELECT saldo INTO saldo_atual FROM public.conta WHERE id = NEW.id_conta;
  IF NEW.valor <= saldo atual THEN
    UPDATE public.conta SET saldo = saldo_atual - NEW.valor WHERE id =
NEW.id conta;
  ELSE
    disponivel := saldo_atual + (SELECT limite FROM public.conta WHERE id =
NEW.id_conta) - (SELECT cheque_especial FROM public.conta WHERE id =
NEW.id conta);
    IF NEW.valor > disponivel THEN
      RAISE EXCEPTION 'Saque não permitido. Cliente não possui limite
disponível.';
    ELSE
      UPDATE conta SET saldo = 0, cheque_especial = cheque_especial +
(NEW.valor - saldo atual) WHERE id = NEW.id conta;
    END IF;
  END IF:
  RETURN NEW;
END:
$verifica_saque$ LANGUAGE plpgsql;
CREATE TRIGGER trigger_verifica_saque
BEFORE INSERT ON public.saque
FOR EACH ROW
EXECUTE FUNCTION verifica_saque();
```

### Testando ela:

## Sacando 800:



Ela ficou com 400 de saldo e com 500 de limite, se eu tentar sacar 1000:



### E se eu tentar sacar 900:

# Query Query History



INSERT 0 1

Query returned successfully in 82 msec.

	id [PK] integer	id_conta integer	valor numeric (15,2)
1	1	1	800.00
2	2	1	900.00

	id [PK] integer	id_cliente integer	saldo numeric (15,2)	limite numeric (15,2)	cheque_especial numeric (15,2)
1	1	1	0.00	500.00	500.00

# Agora, se eu tentar sacar 1, ele vai dar:

# Query Query History 1 INSERT INTO public.saque VALUES (3, 1, 1.00); Data Output Messages Notifications ERROR: Saque não permitido. Cliente não possui limite disponível. CONTEXT: função PL/pgSQL verifica\_saque() linha 14 em RAISE SQL state: P0001