-- Trigger: fc\_reestablece\_valores\_prestamo on public.abonos

-- DROP TRIGGER fc\_reestablece\_valores\_prestamo ON public.abonos;

CREATE TRIGGER fc\_reestablece\_valores\_prestamo

AFTER UPDATE

ON public.abonos

FOR EACH ROW

EXECUTE PROCEDURE public.fc\_reestablece\_valores\_prestamo();

-- Function: public.fc\_reestablece\_valores\_prestamo()
-- DROP FUNCTION public.fc\_reestablece\_valores\_prestamo();
CREATE OR REPLACE FUNCTION public.fc\_reestablece\_valores\_prestamo()
RETURNS trigger AS
$BODY$
DECLARE
s\_query VARCHAR;
result RECORD;
r\_record RECORD;
cteccodigo1 bigint;
cod\_prestamo integer;
f\_prestamo\_id integer;
f\_valor\_prestamo double precision;
f\_tasa double precision;
f\_tipo\_prestamo varchar;
f\_tiempo\_cobro varchar;
f\_cantidad\_cuotas\_pagar integer;
f\_valor\_cuota\_pagar double precision;
f\_fecha\_prestamo date;
f\_fecha\_inicio\_prestamo date;
f\_fecha\_proximo\_cobro date;
f\_valor\_total\_deuda double precision;
f\_valor\_abono\_deuda double precision;
f\_valor\_proximo\_pago\_deuda double precision;
f\_estado varchar;
f\_user\_id integer;
f\_cliente\_id integer;
f\_cuota\_pagada\_abierto integer;
f\_cuota\_pagada\_cerrado integer;
f\_prestamo\_estado\_pago varchar;
BEGIN
IF TG\_OP ='UPDATE' THEN
IF OLD.id IS NOT NULL AND NEW.abono\_estado = 'INACTIVO' THEN
s\_query='SELECT id, prestamo\_valor, prestamo\_tasa, prestamo\_tipo, prestamo\_tiempo\_cobro, prestamo\_numero\_cuotas,
prestamo\_valor\_cuota, prestamo\_fecha, prestamo\_fecha\_inicial, prestamo\_fecha\_proximo\_cobro,
prestamo\_valor\_actual, prestamo\_valor\_abonado, prestamo\_valor\_proxima\_cuota,
prestamo\_estado, user\_id, cliente\_id, abono\_id, created\_at, updated\_at,prestamo\_estado\_pago,
prestamo\_cantidad\_cuota\_pagada\_creditoabierto,prestamo\_cantidad\_cuota\_pagada\_creditocerrado
FROM historico\_prestamos
WHERE abono\_id='||OLD.id||' and id = '||OLD.prestamo\_id||'';
FOR r\_record IN EXECUTE s\_query LOOP
f\_prestamo\_id =r\_record.id;
f\_valor\_prestamo =r\_record.prestamo\_valor;
f\_tasa =r\_record.prestamo\_tasa;
f\_tipo\_prestamo =r\_record.prestamo\_tipo;
f\_tiempo\_cobro =r\_record.prestamo\_tiempo\_cobro;
f\_cantidad\_cuotas\_pagar =r\_record.prestamo\_numero\_cuotas;
f\_valor\_cuota\_pagar =r\_record.prestamo\_valor\_cuota;
f\_fecha\_prestamo =r\_record.prestamo\_fecha;
f\_fecha\_inicio\_prestamo =r\_record.prestamo\_fecha\_inicial;
f\_fecha\_proximo\_cobro =r\_record.prestamo\_fecha\_proximo\_cobro;
f\_valor\_total\_deuda =r\_record.prestamo\_valor\_actual;
f\_valor\_abono\_deuda =r\_record.prestamo\_valor\_abonado;
f\_valor\_proximo\_pago\_deuda =r\_record.prestamo\_valor\_proxima\_cuota;
f\_estado =r\_record.prestamo\_estado;
f\_user\_id =r\_record.user\_id;
f\_cliente\_id =r\_record.cliente\_id;
f\_prestamo\_estado\_pago =r\_record.prestamo\_estado\_pago;
f\_cuota\_pagada\_abierto =r\_record.prestamo\_cantidad\_cuota\_pagada\_creditoabierto;
f\_cuota\_pagada\_cerrado =r\_record.prestamo\_cantidad\_cuota\_pagada\_creditocerrado;
END LOOP;
UPDATE prestamos SET prestamo\_fecha=f\_fecha\_prestamo, prestamo\_fecha\_inicial =f\_fecha\_inicio\_prestamo,
prestamo\_fecha\_proximo\_cobro = f\_fecha\_proximo\_cobro, prestamo\_valor\_actual =f\_valor\_total\_deuda,
prestamo\_valor\_abonado = f\_valor\_abono\_deuda, prestamo\_valor\_proxima\_cuota = f\_valor\_proximo\_pago\_deuda,
prestamo\_estado = f\_estado, prestamo\_estado\_pago = f\_prestamo\_estado\_pago,
prestamo\_cantidad\_cuota\_pagada\_creditoabierto = f\_cuota\_pagada\_abierto,
prestamo\_cantidad\_cuota\_pagada\_creditocerrado = f\_cuota\_pagada\_cerrado
WHERE id = f\_prestamo\_id;
END IF;
END IF;
RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.fc\_reestablece\_valores\_prestamo()
OWNER TO postgres;