-- Trigger: fc\_b\_actualiza\_valores\_prestamo on public.abonos
-- DROP TRIGGER fc\_b\_actualiza\_valores\_prestamo ON public.abonos;
CREATE TRIGGER fc\_b\_actualiza\_valores\_prestamo
AFTER INSERT
ON public.abonos
FOR EACH ROW
EXECUTE PROCEDURE public.fc\_actualiza\_valores\_prestamo();

-- Function: public.fc\_actualiza\_valores\_prestamo()

-- DROP FUNCTION public.fc\_actualiza\_valores\_prestamo();

CREATE OR REPLACE FUNCTION public.fc\_actualiza\_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\_prestamo\_valor double precision; f\_prestamo\_tasa double precision; f\_prestamo\_tipo varchar; f\_prestamo\_tiempo\_cobro varchar; f\_prestamo\_numero\_cuotas integer; f\_prestamo\_valor\_cuota double precision; f\_prestamo\_fecha date; f\_prestamo\_fecha\_inicial date; f\_prestamo\_fecha\_proximo\_cobro date; f\_prestamo\_valor\_actual double precision; f\_prestamo\_valor\_abonado double precision; f\_prestamo\_valor\_proxima\_cuota double precision; f\_prestamo\_estado varchar; f\_prestamo\_estado\_pago varchar; f\_user\_id integer; f\_cliente\_id integer; f\_prestamo\_cantidad\_cuota\_pagada\_creditocerrado integer; f\_prestamo\_cantidad\_cuota\_pagada\_creditoabierto integer; f\_cuotas\_por\_pagar integer; f\_prestamo\_valor\_actual\_nuevo double precision; f\_prestamo\_valor\_abonado\_nuevo double precision; f\_proximovalor\_cancelar double precision; f\_dias\_cobro integer; f\_valor\_total\_cuotapagar\_abierto double precision; f\_valor\_restante\_pago double precision; BEGIN --f\_proximovalor\_cancelar =0; IF TG\_OP ='INSERT' THEN IF NEW.prestamo\_id IS NOT NULL THEN s\_query='SELECT prestamo\_tipo FROM prestamos WHERE id='||NEW.prestamo\_id||' '; FOR r\_record IN EXECUTE s\_query LOOP f\_prestamo\_tipo = r\_record.prestamo\_tipo; END LOOP; /\* INICIO PRESTAMO CERRADO\*/ IF f\_prestamo\_tipo = 'CERRADO' THEN PERFORM fc\_calcula\_pago\_credito\_cerrado(NEW.prestamo\_id, NEW.abono\_valor, NEW.abono\_tipo\_pago); ELSIF f\_prestamo\_tipo = 'ABIERTO' THEN PERFORM fc\_calcula\_pago\_credito\_abierto(NEW.prestamo\_id, NEW.abono\_valor, NEW.abono\_tipo\_pago); END IF; /\* FIN PRESTAMO ABIERTO\*/ END IF; END IF; RETURN NEW; END; $BODY$ LANGUAGE plpgsql VOLATILE COST 100; ALTER FUNCTION public.fc\_actualiza\_valores\_prestamo() OWNER TO postgres;