-- Trigger: fc\_a\_historico\_prestamo on public.abonos

-- DROP TRIGGER fc\_a\_historico\_prestamo ON public.abonos;

CREATE TRIGGER fc\_a\_historico\_prestamo

BEFORE INSERT

ON public.abonos

FOR EACH ROW

EXECUTE PROCEDURE public.fc\_historico\_prestamo();

-- Function: public.fc\_historico\_prestamo()

-- DROP FUNCTION public.fc\_historico\_prestamo();

CREATE OR REPLACE FUNCTION public.fc\_historico\_prestamo()

RETURNS trigger AS

$BODY$

DECLARE

sql VARCHAR;

result RECORD;

result1 RECORD;

cteccodigo1 bigint;

BEGIN

IF TG\_OP ='INSERT' THEN

IF NEW.id IS NOT NULL THEN

INSERT INTO historico\_prestamos (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\_total,

prestamo\_valor\_abonado, prestamo\_valor\_proxima\_cuota, prestamo\_valor\_actual,

prestamo\_estado, user\_id, cliente\_id, abono\_id, created\_at, updated\_at,

prestamo\_cantidad\_cuota\_pagada\_creditocerrado, prestamo\_cantidad\_cuota\_pagada\_creditoabierto,

prestamo\_estado\_pago,prestamo\_observacion, prestamo\_utilidad\_mes,prestamo\_fecha\_utilidad\_mes)

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\_total,

prestamo\_valor\_abonado, prestamo\_valor\_proxima\_cuota, prestamo\_valor\_actual,

prestamo\_estado, user\_id, cliente\_id, NEW.id, created\_at, updated\_at,

prestamo\_cantidad\_cuota\_pagada\_creditocerrado, prestamo\_cantidad\_cuota\_pagada\_creditoabierto,

prestamo\_estado\_pago,prestamo\_observacion, prestamo\_utilidad\_mes,prestamo\_fecha\_utilidad\_mes

FROM prestamos where id=NEW.prestamo\_id;

END IF;

END IF;

RETURN NEW;

END;

$BODY$

LANGUAGE plpgsql VOLATILE

COST 100;

ALTER FUNCTION public.fc\_historico\_prestamo()

OWNER TO postgres;