-- Function: public.fc\_actualiza\_valores\_prestamo\_1(bigint, character varying)

-- DROP FUNCTION public.fc\_actualiza\_valores\_prestamo\_1(bigint, character varying);

CREATE OR REPLACE FUNCTION public.fc\_actualiza\_valores\_prestamo\_1(

bigint,

character varying)

RETURNS bigint AS

$BODY$

DECLARE

f\_prestamo\_id ALIAS FOR $1;

tipo\_pago ALIAS FOR $2;

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\_cuotas\_pagadas\_credito\_cerrado integer;

f\_cuotas\_pagadas\_credito\_abierto integer;

f\_cuotas\_por\_pagar integer;

f\_valor\_total\_deuda\_nuevo double precision;

f\_valor\_abono\_deuda\_nuevo double precision;

f\_proximovalor\_cancelar double precision;

f\_dias\_cobro integer;

f\_valor\_total\_cuotapagar\_abierto double precision;

BEGIN

--IF TG\_OP ='INSERT' THEN

IF f\_prestamo\_id IS NOT NULL THEN

s\_query='SELECT id, valor\_prestamo, tasa, tipo\_prestamo, tiempo\_cobro, cantidad\_cuotas\_pagar,

valor\_cuota\_pagar, fecha\_prestamo, fecha\_inicio\_prestamo, fecha\_proximo\_cobro,

valor\_total\_deuda, valor\_abono\_deuda, valor\_proximo\_pago\_deuda,

estado, user\_id, cliente\_id, created\_at, updated\_at,

CASE WHEN cantidad\_cuota\_pagada\_creditocerrado IS NULL THEN 0

ELSE cantidad\_cuota\_pagada\_creditocerrado END AS cantidad\_cuota\_pagada\_creditocerrado ,

CASE WHEN cantidad\_cuota\_pagada\_creditoabierto IS NULL THEN 0

ELSE cantidad\_cuota\_pagada\_creditoabierto END AS cantidad\_cuota\_pagada\_creditoabierto

FROM prestamos

WHERE id='||f\_prestamo\_id||' ';

raise notice'cantidad%',s\_query;

FOR r\_record IN EXECUTE s\_query LOOP

--f\_prestamo\_id =r\_record.id;

f\_valor\_prestamo =r\_record.valor\_prestamo;

f\_tasa =r\_record.tasa;

f\_tipo\_prestamo =r\_record.tipo\_prestamo;

f\_tiempo\_cobro =r\_record.tiempo\_cobro;

f\_cantidad\_cuotas\_pagar =r\_record.cantidad\_cuotas\_pagar;

f\_valor\_cuota\_pagar =r\_record.valor\_cuota\_pagar;

f\_fecha\_prestamo =r\_record.fecha\_prestamo;

f\_fecha\_inicio\_prestamo =r\_record.fecha\_inicio\_prestamo;

f\_fecha\_proximo\_cobro =r\_record.fecha\_proximo\_cobro;

f\_valor\_total\_deuda =r\_record.valor\_total\_deuda;

f\_valor\_abono\_deuda =r\_record.valor\_abono\_deuda;

f\_valor\_proximo\_pago\_deuda =r\_record.valor\_proximo\_pago\_deuda;

f\_estado =r\_record.estado;

f\_user\_id =r\_record.user\_id;

f\_cliente\_id =r\_record.cliente\_id;

f\_cuotas\_pagadas\_credito\_cerrado =r\_record.cantidad\_cuota\_pagada\_creditocerrado;

f\_cuotas\_pagadas\_credito\_abierto =r\_record.cantidad\_cuota\_pagada\_creditoabierto;

/\* SE CALCULA EN NUMERO LA CANTIDAD DE DIAS A SUMAR A LA PROFIMA FECHA DE PAGO\*/

IF f\_tiempo\_cobro = 'DIARIO' THEN

f\_dias\_cobro = 1;

ELSIF f\_tiempo\_cobro = 'SEMANAL' THEN

f\_dias\_cobro = 7;

ELSIF f\_tiempo\_cobro = 'QUINCENAL' THEN

f\_dias\_cobro = 15;

ELSIF f\_tiempo\_cobro = 'MENSUAL' THEN

f\_dias\_cobro = 30;

END IF;

/\* INICIO PRESTAMO CERRADO\*/

IF f\_tipo\_prestamo = 'CERRADO' THEN

f\_valor\_total\_deuda\_nuevo = f\_valor\_total\_deuda - 15000 ;

f\_valor\_abono\_deuda\_nuevo = f\_valor\_abono\_deuda + 15000;

IF tipo\_pago ='PAGO' THEN

-- SE CALCULA LAS CUOTAS RESTANTES PARA DIVIDIR

f\_cuotas\_pagadas\_credito\_cerrado = f\_cuotas\_pagadas\_credito\_cerrado + 1;

f\_cuotas\_por\_pagar = (f\_cantidad\_cuotas\_pagar - f\_cuotas\_pagadas\_credito\_cerrado);

IF 15000 <= f\_valor\_proximo\_pago\_deuda THEN

f\_proximovalor\_cancelar = f\_valor\_proximo\_pago\_deuda + (f\_valor\_proximo\_pago\_deuda - 15000);

ELSIF 15000 > f\_valor\_proximo\_pago\_deuda THEN

f\_proximovalor\_cancelar = f\_valor\_total\_deuda\_nuevo / f\_cuotas\_por\_pagar;

END IF;

ELSIF tipo\_pago ='ABONO A CAPITAL' THEN

f\_cuotas\_pagadas\_credito\_cerrado = f\_cuotas\_pagadas\_credito\_cerrado ;

/\* si realiza abono y se resta la cuota a pagar menos el valor a pagar \*/

IF 15000 <= f\_valor\_proximo\_pago\_deuda THEN

f\_proximovalor\_cancelar = f\_valor\_proximo\_pago\_deuda - 15000;

END IF;

END IF;

/\* FIN PRESTAMO CERRADO\*/

/\* INICIO PRESTAMO ABIERTO\*/

ELSIF f\_tipo\_prestamo = 'ABIERTO' THEN

IF tipo\_pago ='PAGO' THEN

-- SE CALCULA LAS CUOTAS RESTANTES PARA DIVIDIR

f\_cuotas\_pagadas\_credito\_cerrado = f\_cuotas\_pagadas\_credito\_cerrado + 1;

f\_cuotas\_por\_pagar = (f\_cantidad\_cuotas\_pagar - f\_cuotas\_pagadas\_credito\_cerrado);

/\* se calcula el valor total de la cuota a pagar en caso de que el pago sea quincenal\*/

f\_valor\_total\_cuotapagar\_abierto = f\_cantidad\_cuotas\_pagar \* f\_valor\_proximo\_pago\_deuda;

IF 15000 <= f\_valor\_proximo\_pago\_deuda THEN

f\_proximovalor\_cancelar = f\_valor\_proximo\_pago\_deuda + (f\_valor\_proximo\_pago\_deuda - 15000);

ELSIF 15000 > f\_valor\_proximo\_pago\_deuda THEN

IF f\_cantidad\_cuotas\_pagar = 2 AND f\_cuotas\_pagadas\_credito\_cerrado = 1 THEN

/\* se calcula si la primera cuota pagada supera el valor total de la cuota mensual o no\*/

IF 15000 > f\_valor\_total\_cuotapagar\_abierto THEN

/\* se calcula el nuevo valor total y valor abono de la deuda\*/

f\_valor\_total\_deuda\_nuevo = (f\_valor\_total\_deuda - (15000 - f\_valor\_total\_cuotapagar\_abierto)) ;

f\_valor\_abono\_deuda\_nuevo = (f\_valor\_abono\_deuda + (15000- f\_valor\_total\_cuotapagar\_abierto));

/\* se calcula el nuevo interes\*/

f\_proximovalor\_cancelar = (((f\_valor\_total\_deuda\_nuevo \* f\_tasa)/100)/f\_cantidad\_cuotas\_pagar);

/\* se multiplica los dias por 2 ya que se le sumaria un mes completo a el pago\*/

f\_dias\_cobro = f\_dias\_cobro \* 2;

ELSIF 15000 < f\_valor\_total\_cuotapagar\_abierto THEN

/\* se calcula el valor de la segunda cuota \*/

f\_proximovalor\_cancelar = f\_valor\_total\_cuotapagar\_abierto - 15000;

END IF;

ELSIF f\_cantidad\_cuotas\_pagar = 2 AND f\_cuotas\_pagadas\_credito\_cerrado = 2 THEN

/\* se calcula si la segunda cuota pagada supera el valor de la cuota a pagar\*/

IF 15000 > f\_valor\_proximo\_pago\_deuda THEN

/\* se calcula el nuevo valor total y valor abono de la deuda\*/

f\_valor\_total\_deuda\_nuevo = (f\_valor\_total\_deuda - (15000 - f\_valor\_proximo\_pago\_deuda)) ;

f\_valor\_abono\_deuda\_nuevo = (f\_valor\_abono\_deuda + (15000- f\_valor\_proximo\_pago\_deuda));

/\* se calcula el nuevo interes\*/

f\_proximovalor\_cancelar = (((f\_valor\_total\_deuda\_nuevo \* f\_tasa)/100)/f\_cantidad\_cuotas\_pagar);

ELSIF 15000 <= f\_valor\_proximo\_pago\_deuda THEN

/\* se calcula el valor de la segunda cuota \*/

f\_proximovalor\_cancelar = f\_valor\_proximo\_pago\_deuda - 15000;

END IF;

END IF;

END IF;

ELSIF tipo\_pago ='ABONO A CAPITAL' THEN

/\* se calcula el nuevo valor total y valor abono de la deuda\*/

f\_valor\_total\_deuda\_nuevo = (f\_valor\_total\_deuda - (15000 - f\_valor\_total\_cuotapagar\_abierto)) ;

f\_valor\_abono\_deuda\_nuevo = (f\_valor\_abono\_deuda + (15000- f\_valor\_total\_cuotapagar\_abierto));

/\* se calcula el nuevo interes\*/

f\_proximovalor\_cancelar = (((f\_valor\_total\_deuda\_nuevo \* f\_tasa)/100)/f\_cantidad\_cuotas\_pagar);

END IF;

END IF;

/\* FIN PRESTAMO ABIERTO\*/

END LOOP;

UPDATE prestamos SET valor\_total\_deuda = f\_valor\_total\_deuda\_nuevo, valor\_abono\_deuda=f\_valor\_abono\_deuda\_nuevo,

cantidad\_cuota\_pagada\_creditocerrado =f\_cuotas\_pagadas\_credito\_cerrado,

valor\_proximo\_pago\_deuda=f\_proximovalor\_cancelar,

fecha\_proximo\_cobro = (SELECT f\_fecha\_proximo\_cobro::DATE + f\_dias\_cobro)

WHERE id = f\_prestamo\_id;

--UPDATE prestamos SET valor\_abono\_deuda=70000 WHERE id=NEW.prestamo\_id;

END IF;

-- END IF;

RETURN f\_valor\_total\_deuda\_nuevo;

END;

$BODY$

LANGUAGE plpgsql VOLATILE

COST 100;

ALTER FUNCTION public.fc\_actualiza\_valores\_prestamo\_1(bigint, character varying)

OWNER TO postgres;