-- Function: public.fc\_calcula\_utilidad\_mes(bigint)
-- DROP FUNCTION public.fc\_calcula\_utilidad\_mes(bigint);
CREATE OR REPLACE FUNCTION public.fc\_calcula\_utilidad\_mes(bigint)
RETURNS bigint AS
$BODY$
DECLARE
f\_prestamo\_id ALIAS FOR $1;
s\_query VARCHAR;
result RECORD;
r\_record RECORD;
cteccodigo1 bigint;
cod\_prestamo integer;
f\_tipo\_prestamo varchar;
f\_tiempo\_cobro varchar;
f\_cantidad\_cuotas\_pagar integer;
f\_valor\_cuota\_pagar double precision;
f\_fecha\_utilidad\_mes date;
f\_fecha\_inicio\_prestamo date;
f\_fecha\_proximo\_cobro date;
f\_valor\_proximo\_pago\_deuda double precision;
f\_proximovalor\_cancelar double precision;
f\_bandera\_calcula\_utilidad\_mes integer;
f\_cantidad\_dias\_primera\_cuota integer;
f\_dias integer;
f\_cuotas\_restantes\_mes integer;
f\_dias\_multiplicar\_cuota integer;
f\_utilidad\_mes double precision;
f\_valor\_utilidad\_mes double precision;
f\_prestamo\_interes\_cerrado double precision;
f\_prestamo\_interes\_abierto double precision;
f\_cuotas\_pagar\_primer\_mes integer;
f\_cuotas\_pagar\_total integer;
f\_cuotas\_por\_pagar\_cerrado integer;
f\_prestamo\_valor\_proxima\_cuota double precision;
f\_prestamo\_estado\_pago varchar;
BEGIN
--IF TG\_OP ='INSERT' THEN
IF f\_prestamo\_id IS NOT NULL THEN
s\_query='SELECT id, prestamo\_tipo, prestamo\_tiempo\_cobro, prestamo\_numero\_cuotas,
prestamo\_valor\_cuota, prestamo\_fecha\_inicial, prestamo\_fecha\_proximo\_cobro,
prestamo\_valor\_proxima\_cuota, prestamo\_utilidad\_mes, prestamo\_estado,
CASE WHEN prestamo\_fecha\_proximo\_cobro > prestamo\_fecha\_utilidad\_mes THEN 1
ELSE 2 END AS bandera\_calcula\_utilidad\_mes,
(select substring((select (date\_trunc(''month'',prestamo\_fecha\_proximo\_cobro) +''1month'' ::interval -''1sec'' ::interval)::varchar),1,10) )
as fecha\_utilidad\_mes,
(select substring((select (date\_trunc(''month'',current\_date)
+''1month'' ::interval -''1sec''::interval)::varchar),1,10)::date - prestamo\_fecha\_inicial ) as cantidad\_dias\_primera\_cuota,
((prestamo\_valor \* prestamo\_tasa)/100) as prestamo\_interes\_cerrado,
((prestamo\_valor\_actual \* prestamo\_tasa)/100) as prestamo\_interes\_abierto ,
(prestamo\_numero\_cuotas - prestamo\_cantidad\_cuota\_pagada\_creditocerrado) as cuotas\_por\_pagar\_cerrado,
prestamo\_valor\_proxima\_cuota, prestamo\_estado\_pago
FROM prestamos
WHERE id='||f\_prestamo\_id||' and prestamo\_estado = ''ACTIVO'' ';
raise notice'cantidad%',s\_query;
FOR r\_record IN EXECUTE s\_query LOOP
--f\_prestamo\_id =r\_record.id;
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\_inicio\_prestamo =r\_record.prestamo\_fecha\_inicial;
f\_fecha\_proximo\_cobro =r\_record.prestamo\_fecha\_proximo\_cobro;
f\_valor\_proximo\_pago\_deuda =r\_record.prestamo\_valor\_proxima\_cuota;
f\_bandera\_calcula\_utilidad\_mes =r\_record.bandera\_calcula\_utilidad\_mes;
f\_fecha\_utilidad\_mes =r\_record.fecha\_utilidad\_mes;
f\_cantidad\_dias\_primera\_cuota =r\_record.cantidad\_dias\_primera\_cuota;
f\_valor\_utilidad\_mes =r\_record.prestamo\_utilidad\_mes;
f\_prestamo\_interes\_cerrado =r\_record.prestamo\_interes\_cerrado;
f\_prestamo\_interes\_abierto =r\_record.prestamo\_interes\_abierto;
f\_cuotas\_por\_pagar\_cerrado =r\_record.cuotas\_por\_pagar\_cerrado;
f\_prestamo\_valor\_proxima\_cuota =r\_record.prestamo\_valor\_proxima\_cuota;
f\_prestamo\_estado\_pago =r\_record.prestamo\_estado\_pago;
END LOOP;
/\*si la utilidad es cero quiere decir que es la primera cuota a pagar \*/
IF f\_valor\_utilidad\_mes = 0 THEN
f\_bandera\_calcula\_utilidad\_mes = 1;
END IF;
IF f\_valor\_utilidad\_mes IS NULL OR f\_bandera\_calcula\_utilidad\_mes = 1 THEN
IF f\_tiempo\_cobro = 'SEMANAL' THEN
f\_dias = 7;
f\_dias\_multiplicar\_cuota = 4;
ELSIF f\_tiempo\_cobro = 'QUINCENAL' THEN
f\_dias = 15;
f\_dias\_multiplicar\_cuota = 2;
ELSIF f\_tiempo\_cobro = 'MENSUAL' THEN
f\_dias = 30;
f\_dias\_multiplicar\_cuota = 1;
END IF;
IF f\_tipo\_prestamo = 'ABIERTO' THEN
/\* si es la primera cuota a pagar \*/
f\_cuotas\_restantes\_mes = floor(f\_cantidad\_dias\_primera\_cuota / f\_dias);
raise notice'cuota%',f\_cantidad\_dias\_primera\_cuota;
raise notice'dias%',f\_dias;
/\* se calculan las primeras cuotas del mes a pagar para cuando se realizen
los primeros abonos se puedan calcular bien los valores este calculo solo aplica
para el credito abiero\*/
f\_cuotas\_pagar\_primer\_mes = f\_cuotas\_restantes\_mes + 1;
IF f\_fecha\_inicio\_prestamo = f\_fecha\_proximo\_cobro THEN
raise notice'fechas iguales%',f\_cuotas\_restantes\_mes;
IF f\_cuotas\_restantes\_mes >= 1 THEN
/\* se determina cuantas cuotas a pagar tendra la persona en este mes\*/
f\_utilidad\_mes = ( f\_valor\_proximo\_pago\_deuda + (f\_valor\_cuota\_pagar \* f\_cuotas\_restantes\_mes));
ELSIF f\_cuotas\_restantes\_mes < 1 THEN
/\* si la primera cuota se paga en el proximo mes, el valor de la utilidad en este mes es 0 \*/
--f\_utilidad\_mes = 0;
f\_utilidad\_mes = ( f\_valor\_proximo\_pago\_deuda + (f\_valor\_cuota\_pagar \* f\_cuotas\_restantes\_mes));
END IF;
ELSIF f\_fecha\_inicio\_prestamo < f\_fecha\_proximo\_cobro THEN
f\_utilidad\_mes = f\_prestamo\_interes\_abierto;
END IF;
f\_cuotas\_pagar\_total = f\_cuotas\_pagar\_primer\_mes;
ELSIF f\_tipo\_prestamo = 'CERRADO' THEN
/\* la utilidad aqui se calcula tomando el interes general del credito y dividiendolo por la cuotas que falta pagar en el mes, mas la primera cuota
siempre y cuando el numero de cuotas restantes del mes sea menor a el numero de cuotas a pagar \*/
/\* si es la primera cuota a pagar \*/
f\_cuotas\_restantes\_mes = floor(f\_cantidad\_dias\_primera\_cuota / f\_dias);
/\* se calculan las primeras cuotas del mes a pagar para cuando se realizen
los primeros abonos se puedan calcular bien los valores este calculo solo aplica
para el credito cerrado\*/
f\_cuotas\_pagar\_primer\_mes = f\_cuotas\_restantes\_mes + 1;
IF f\_fecha\_inicio\_prestamo = f\_fecha\_proximo\_cobro THEN
/\* si es el primer calculo de la utilidad se le suma 1 que seria la primera cuota,
si no es el primer calculo, no se le suma\*/
IF f\_cuotas\_pagar\_primer\_mes < 1 THEN
/\* si la primera cuota se paga en el proximo mes, el valor de la utilidad en este mes es 0 \*/
f\_utilidad\_mes = 0;
ELSIF f\_cuotas\_pagar\_primer\_mes >= 1 THEN
f\_utilidad\_mes = ROUND((f\_prestamo\_interes\_cerrado / f\_cantidad\_cuotas\_pagar) \* (f\_cuotas\_restantes\_mes + 1)); END IF;
ELSE
f\_utilidad\_mes = ROUND((f\_prestamo\_interes\_cerrado / f\_cantidad\_cuotas\_pagar) \* f\_cuotas\_restantes\_mes);
END IF;
ELSIF f\_fecha\_inicio\_prestamo < f\_fecha\_proximo\_cobro THEN
/\* se calculan las cuotas restantes en el mes teniendo encuenta que hay un limete de cuotas ,si las cuotas restantes del mes,
superan las cuotas a pagar ,se resta por la cuota del mes\*/
f\_utilidad\_mes = ROUND((f\_prestamo\_interes\_cerrado / f\_cantidad\_cuotas\_pagar) \* f\_cuotas\_por\_pagar\_cerrado);
END IF;
f\_cuotas\_pagar\_total = f\_cantidad\_cuotas\_pagar;
/\* se verifica antes de totalizar si el prestmo biene en color (morado o rojo) ya que si biene en estos colores
la utilidad es mayor,por el excedente del valor que trae la cuota anterior \*/
IF f\_prestamo\_estado\_pago = 'MORADO' OR f\_prestamo\_estado\_pago = 'ROJO' THEN
/\* se le quita una cuota al total de la utilidad ya que esta sera reemplazada por el valor\_proxima cuota\*/
f\_utilidad\_mes = (f\_prestamo\_valor\_proxima\_cuota + (f\_utilidad\_mes - (f\_utilidad\_mes /f\_dias\_multiplicar\_cuota)));
END IF;
/\* se llama la funcion qe totalice\*/
PERFORM fc\_utilidad\_total\_prestamo\_mes(f\_utilidad\_mes, f\_fecha\_utilidad\_mes);
/\* se actualiza el valor de la utilidad del mes y el valor mensual \*/
UPDATE prestamos SET prestamo\_utilidad\_mes = f\_utilidad\_mes,
prestamo\_fecha\_utilidad\_mes = f\_fecha\_utilidad\_mes,
prestamo\_numero\_cuotas = f\_cuotas\_pagar\_total
WHERE id = f\_prestamo\_id;
END IF;
END IF;
-- END IF;
RETURN 1;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.fc\_calcula\_utilidad\_mes(bigint)
OWNER TO postgres;