-- Function: public.fc\_utilidad\_prestamo(bigint, character varying)
-- DROP FUNCTION public.fc\_utilidad\_prestamo(bigint, character varying);
CREATE OR REPLACE FUNCTION public.fc\_utilidad\_prestamo(
bigint,
character varying)
RETURNS double precision AS
$BODY$
DECLARE
s\_cobro ALIAS FOR $1;
s\_tipo\_valor ALIAS FOR $2;
sql VARCHAR;
s\_query VARCHAR;
f\_tiempo\_prestamo VARCHAR;
f\_abono\_tipo\_pago VARCHAR;
result RECORD;
result1 RECORD;
r\_record RECORD;
cod\_prestamo integer;
f\_valor\_real\_pagar double precision;
f\_tasa double precision;
f\_tasa\_sin\_puntos double precision;
f\_tasa\_formula double precision;
f\_tasa\_decimal double precision;
f\_valor\_capital\_recaudado double precision;
f\_valor\_interes\_recaudado double precision;
f\_valor\_retornar double precision;
f\_abono\_capital\_abonado\_abierto double precision;
BEGIN
-- RAISE NOTICE 'cobro-- %',s\_cobro;
IF s\_cobro IS NOT NULL THEN
s\_query='SELECT abono\_valor,prestamo\_tasa,replace(prestamo\_tasa::text,''.'','''')::double precision as tasa\_decimal,
prestamo\_tipo,abono\_tipo\_pago,abono\_capital\_abonado\_abierto
FROM abonos c ,prestamos p WHERE c.prestamo\_id = p.id and c.id='||s\_cobro||' ';
raise notice'cantidad%',s\_query;
FOR r\_record IN EXECUTE s\_query LOOP
f\_valor\_real\_pagar = r\_record.abono\_valor;
f\_tasa = r\_record.prestamo\_tasa;
f\_tasa\_decimal = r\_record.tasa\_decimal;
f\_tiempo\_prestamo = r\_record.prestamo\_tipo;
f\_abono\_tipo\_pago = r\_record.abono\_tipo\_pago;
f\_abono\_capital\_abonado\_abierto = r\_record.abono\_capital\_abonado\_abierto;
--raise notice'valor real a pagar%',f\_valor\_real\_pagar;
raise notice'tasa%',f\_tasa\_decimal;
--RAISE NOTICE 'clasesss-- %',f\_prestamo\_id;
IF f\_tasa > 9 THEN
f\_tasa\_sin\_puntos = replace(f\_tasa::text,'.'::text,''::text)::double precision;
f\_tasa\_formula = 1||'.'||f\_tasa\_sin\_puntos||'0';
ELSIF f\_tasa <= 9 THEN
f\_tasa\_sin\_puntos = replace(f\_tasa::text,'.'::text,''::text)::double precision;
f\_tasa\_formula = 1||'.0'||f\_tasa\_sin\_puntos;
END IF;
raise notice'tasa formula%',f\_tasa\_formula;
raise notice'tasa formula%',f\_tasa\_formula;
f\_valor\_capital\_recaudado = round(f\_valor\_real\_pagar / f\_tasa\_formula);
f\_valor\_interes\_recaudado = round((f\_tasa \* f\_valor\_capital\_recaudado)/100);
END LOOP;
END IF;
IF f\_tiempo\_prestamo = 'CERRADO' THEN
IF s\_tipo\_valor = 'CAPITAL' THEN
f\_valor\_retornar = f\_valor\_capital\_recaudado ;
ELSIF s\_tipo\_valor = 'INTERES' THEN
f\_valor\_retornar = f\_valor\_interes\_recaudado ;
END IF;
ELSIF f\_tiempo\_prestamo = 'ABIERTO' THEN
IF f\_abono\_tipo\_pago = 'PAGO CUOTA' AND f\_abono\_capital\_abonado\_abierto > 0 THEN
IF s\_tipo\_valor = 'CAPITAL' THEN
f\_valor\_retornar = f\_abono\_capital\_abonado\_abierto;
ELSIF s\_tipo\_valor = 'INTERES' THEN
f\_valor\_retornar = round(f\_valor\_real\_pagar - f\_abono\_capital\_abonado\_abierto) ;
END IF;
ELSIF f\_abono\_tipo\_pago = 'PAGO CUOTA' AND f\_abono\_capital\_abonado\_abierto <= 0 THEN
IF s\_tipo\_valor = 'CAPITAL' THEN
f\_valor\_retornar = 0;
ELSIF s\_tipo\_valor = 'INTERES' THEN
f\_valor\_retornar = f\_valor\_real\_pagar ;
END IF;
ELSIF f\_abono\_tipo\_pago = 'ABONO A CAPITAL' OR f\_abono\_tipo\_pago = 'PAGO TOTAL DEUDA' THEN
IF s\_tipo\_valor = 'CAPITAL' THEN
f\_valor\_retornar = f\_valor\_real\_pagar;
ELSIF s\_tipo\_valor = 'INTERES' THEN
f\_valor\_retornar = 0 ;
END IF;
END IF;
END IF;
-- END IF;
RETURN f\_valor\_retornar;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.fc\_utilidad\_prestamo(bigint, character varying)
OWNER TO postgres;