-- Function: public.fc\_calcula\_fecha\_cobro(bigint, character varying, date)
-- DROP FUNCTION public.fc\_calcula\_fecha\_cobro(bigint, character varying, date);
CREATE OR REPLACE FUNCTION public.fc\_calcula\_fecha\_cobro(
bigint,
character varying,
date)
RETURNS date AS
$BODY$
DECLARE
f\_prestamo\_id ALIAS FOR $1;
f\_tiempo\_cobro\_prestamo ALIAS FOR $2;
f\_fecha\_proximo\_cobro ALIAS FOR $3;
s\_query VARCHAR;
result RECORD;
r\_record RECORD;
cteccodigo1 bigint;
cod\_prestamo integer;
f\_dia\_inicio bigint;
f\_tipo\_prestamo VARCHAR;
f\_cuotas\_pagadas\_credito\_cerrado bigint;
f\_cuotas\_pagadas\_credito\_abierto bigint;
f\_cantidad\_cuotas\_pagadas bigint;
f\_dia VARCHAR;
f\_fecha\_proximo\_pago date;
BEGIN
--IF TG\_OP ='INSERT' THEN
s\_query='SELECT extract(DAY FROM prestamo\_fecha\_inicial) AS dia\_inicio, prestamo\_tipo,
CASE WHEN prestamo\_cantidad\_cuota\_pagada\_creditocerrado IS NULL THEN 0
ELSE prestamo\_cantidad\_cuota\_pagada\_creditocerrado END AS prestamo\_cantidad\_cuota\_pagada\_creditocerrado ,
CASE WHEN prestamo\_cantidad\_cuota\_pagada\_creditoabierto IS NULL THEN 0
ELSE prestamo\_cantidad\_cuota\_pagada\_creditoabierto END AS prestamo\_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\_dia\_inicio =r\_record.dia\_inicio;
f\_tipo\_prestamo =r\_record.prestamo\_tipo;
f\_cuotas\_pagadas\_credito\_cerrado =r\_record.prestamo\_cantidad\_cuota\_pagada\_creditocerrado;
f\_cuotas\_pagadas\_credito\_abierto =r\_record.prestamo\_cantidad\_cuota\_pagada\_creditoabierto;
END LOOP;
IF f\_tipo\_prestamo = 'ABIERTO' THEN
f\_cantidad\_cuotas\_pagadas = f\_cuotas\_pagadas\_credito\_abierto;
ELSIF f\_tipo\_prestamo = 'CERRADO' THEN
f\_cantidad\_cuotas\_pagadas = f\_cuotas\_pagadas\_credito\_cerrado;
END IF;
IF f\_fecha\_proximo\_cobro IS NOT NULL THEN
IF f\_tiempo\_cobro\_prestamo = 'MENSUAL' THEN
s\_query=' SELECT CASE WHEN '''||f\_fecha\_proximo\_cobro||'''::DATE = (select substring((select (date\_trunc(''month'',('''||f\_fecha\_proximo\_cobro||'''::date )) +''1month'' ::interval -''1sec'' ::interval)::varchar),1,10) )::date THEN
(select substring((select (date\_trunc(''month'',('''||f\_fecha\_proximo\_cobro||'''::date +''1month'' ::interval)) +''1month'' ::interval -''1sec'' ::interval)::varchar),1,10) )::DATE
ELSE substring(('''||f\_fecha\_proximo\_cobro||'''::DATE +''1month'' ::interval)::text,1,10)::DATE END AS fecha\_cobro ';
--raise notice'cantidad%',s\_query;
/\* se calcula si el prestamo es quincenal y si se esta pagando la segunda cuota\*/
ELSIF f\_tiempo\_cobro\_prestamo = 'QUINCENAL' AND f\_cantidad\_cuotas\_pagadas = 1 THEN
/\* se calcula si el inicio del prestamo fue el primer dia del mes, para que en la segunda cuota a pagar traiga el dia final del mes \*/
IF f\_dia\_inicio = 1 THEN
s\_query=' select substring((select (date\_trunc(''month'',('''||f\_fecha\_proximo\_cobro||'''::date )) +''1month'' ::interval -''1sec'' ::interval)::varchar),1,10)::date AS fecha\_cobro ';
ELSIF f\_dia\_inicio > 1 THEN
/\* si el dia de inicio del prestamo es diferente del primer dia,
se calcula para que la fecha calculada obtenga tambien ese dia y asi las fechas no se corran\*/
IF f\_dia\_inicio < 10 THEN
f\_dia = '0'''|| f\_dia\_inicio ||'''';
ELSE
f\_dia = f\_dia\_inicio;
END IF;
s\_query='SELECT (extract(YEAR FROM fecha\_calculada) ||'''||'-'||'''|| extract(MONTH FROM fecha\_calculada) ||'''||'-'||'''|| '''||f\_dia||''') AS fecha\_cobro
FROM
( SELECT substring(('''||f\_fecha\_proximo\_cobro||'''::DATE +''15 days'' ::interval)::text,1,10)::DATE AS fecha\_calculada) AS TABLA ';
raise notice'consultaaaa%',s\_query;
END IF;
ELSIF f\_tiempo\_cobro\_prestamo = 'QUINCENAL' THEN
s\_query='SELECT substring(('''||f\_fecha\_proximo\_cobro||'''::DATE +''15 days'' ::interval)::text,1,10)::DATE AS fecha\_cobro ';
ELSIF f\_tiempo\_cobro\_prestamo = 'SEMANAL' THEN
s\_query='SELECT substring(('''||f\_fecha\_proximo\_cobro||'''::DATE +''7 days'' ::interval)::text,1,10)::DATE AS fecha\_cobro ';
END IF;
FOR r\_record IN EXECUTE s\_query LOOP
f\_fecha\_proximo\_pago =r\_record.fecha\_cobro;
END LOOP;
END IF;
RETURN f\_fecha\_proximo\_pago;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION public.fc\_calcula\_fecha\_cobro(bigint, character varying, date)
OWNER TO postgres;