TRIGGER REGISTRAR PERSONA CON TELEFONO INTELIGENTE

CREATE OR REPLACE FUNCTION nueva\_persona()

RETURNS TRIGGER AS $$

BEGIN

INSERT INTO persona (mac\_add)

VALUES (NEW.mac\_add) ON CONFLICT DO NOTHING;

return NEW;

END

$$ LANGUAGE 'plpgsql'

CREATE TRIGGER nuevo\_ingreso\_cc

BEFORE INSERT ON log\_cc

FOR EACH ROW

EXECUTE PROCEDURE nueva\_persona();

TRIGGER VALIDAR CLIENTE DENTRO DEL CC ANTES DE REGISTRAR VENTA

CREATE OR REPLACE FUNCTION validar\_venta()

RETURNS TRIGGER AS $$

DECLARE

ni int;

BEGIN

IF NEW.mac\_add != '' THEN

SELECT sum(count) INTO ni FROM log\_cc where mac\_add = NEW.mac\_add;

IF NOT FOUND THEN

RAISE EXCEPTION 'No existe ese cliente';

END IF;

IF NI > 0 THEN

return NEW;

ELSE

RAISE EXCEPTION 'Cliente no esta presente en el cc';

END IF;

ELSE

return NEW;

END IF;

END

$$ LANGUAGE 'plpgsql'

CREATE TRIGGER venta\_valida

BEFORE INSERT ON ventas

FOR EACH ROW

EXECUTE PROCEDURE validar\_venta();

TRIGGER MARCAR MESA COMO USADA LUEGO DE LIBERARSE

CREATE OR REPLACE FUNCTION mesa\_sucia()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.estado = 0 THEN

UPDATE mesa SET limpiar = 1

WHERE id = NEW.id\_mesa;

END IF;

return NEW;

END

$$ LANGUAGE 'plpgsql'

CREATE TRIGGER mesa\_usada

BEFORE INSERT ON mesa\_log

FOR EACH ROW

EXECUTE PROCEDURE mesa\_sucia();

PA CALCULAR EL PAGO DE ALQUILER DE CADA LOCAL COMO EL 10% DE SUS VENTAS

CREATE OR REPLACE FUNCTION pago\_alquiler(integer) RETURNS integer AS $$

DECLARE

local\_id ALIAS FOR $1;

resultado INTEGER;

ni int;

BEGIN

SELECT

sum(monto)

INTO

ni

FROM

ventas

WHERE

id\_tienda = local\_id AND ventas.time\_stamp > (CURRENT\_DATE - '1 mon'::interval);

resultado := ni\*0.1;

RETURN resultado;

END;

$$ LANGUAGE plpgsql;

Ejemplo:

SELECT pago\_alquiler(tienda\_id);

PA EVALUAR SI TODAS LAS MESAS ESTAN SUCIAS PARA ENVIAR PERSONAL DE LIMPIEZA

CREATE OR REPLACE FUNCTION getLimpiarMesas() RETURNS integer AS $$

DECLARE

ni int;

BEGIN

select count(limpiar) INTO ni from mesa where limpiar = 0;

IF NOT FOUND THEN

RETURN 1;

END IF;

IF NI > 0 THEN

RETURN 0;

ELSE

RETURN 1;

END IF;

END;

$$ LANGUAGE plpgsql;

Ejemplo:

SELECT getLimpiarMesas();

PA LIMPIAR TODAS LAS MESAS

CREATE OR REPLACE FUNCTION LimpiarMesas() RETURNS integer AS $$

BEGIN

UPDATE mesa SET limpiar = 0;

return 1;

END;

$$ LANGUAGE plpgsql;

Ejemplo:

SELECT LimpiarMesas();

PA ALMACENAR HORA DE ENTRADA CC EN VENTA

CREATE OR REPLACE FUNCTION nueva\_compra(integer,integer,integer) RETURNS text AS $$

DECLARE

tienda\_id\_e ALIAS FOR $1;

cedula\_e ALIAS FOR $2;

monto\_e ALIAS FOR $3;

resultado INTEGER;

n2 TEXT;

n3 timestamp;

BEGIN

select mac\_add INTO n2 from persona where cedula = cedula\_e;

select time\_stamp INTO n3 from log\_cc WHERE mac\_add = n2 and count = 1 ORDER BY ID DESC LIMIT 1;

INSERT INTO ventas (id\_tienda,monto,mac\_add,hora\_entrada) VALUES (tienda\_id\_e, monto\_e, n2, n3);

RETURN n2;

END;

$$ LANGUAGE plpgsql;

Ejemplo

select nueva\_compra(1,21291715,300)