***Proyecto Facturador***

****

* **Alumno: Ayala Kevin Elias**
* **Tutor: Cabana Fernando**
* **Profesor: Redondo Camilo**
* **Comisión: 40750**

***El proyecto detallado a continuación tiene la función de facturar con el nombre de la persona, recibo, producto, ciudad, cantidad y la garantía del producto comprado***

***Creación de las tablas***

***y***

***base de datos***

* ***En la siguiente línea de código, creamos la base de datos sobre la cual trabajaremos.***

*Créate database facturador;*

* + ***Creamos la tabla ‘CLIENTE’, dónde se guardará los datos del mismo.***
* *DROP TABLE IF EXISTS Cliente;*
* *CREATE TABLE Cliente (*
* *ID\_Cliente int NOT NULL PRIMARY KEY AUTO\_INCREMENT,*
* *Nombre varchar (600) NOT NULL,*
* *Apellido varchar (600) NOT NULL,*
* *Mail varchar (600) NOT NULL UNIQUE,*
* *Dni int UNSIGNED NOT NULL UNIQUE,*
* *Provincia varchar (1000) NOT NULL,*
* *Localidad varchar (1000) NOT NULL,*
* *Telefono int UNSIGNED NOT NULL UNIQUE*
* *);*
  + ***Creamos la tabla ‘PROVINCIA’, dónde se consultará a cual pertenece nuestro cliente00000000000000000000.***
* *DROP TABLE IF EXISTS Provincias;*
* *CREATE TABLE Provincias (*
* *ID\_Provincias int NOT NULL PRIMARY KEY AUTO\_INCREMENT,*
* *Provincia varchar (600) NOT NULL);*
  + ***Creamos la tabla ‘LOCALIDADES’ para poder consultar a qué provincia y que localidad pertenece nuestro cliente***

***DROP TABLE IF EXISTS Localidad;***

***CREATE TABLE Localidad (ID\_Localidad int NOT NULL AUTO\_INCREMENT PRIMARY KEY,***

***Provincias\_ID int NOT NULL,***

***Localidad varchar (600) NOT NULL );***

* + ***Creamos la tabla ‘PRODUCTOS’ dónde se guardarán los datos ingresados a nuestra base de datos.***

***DROP TABLE IF EXISTS Productos;***

***CREATE TABLE Productos (***

***ID\_Producto int AUTO\_INCREMENT NOT NULL PRIMARY KEY,***

***Marca varchar (600) NOT NULL,***

***Modelo varchar (600) NOT NULL,***

***Precio int UNSIGNED NOT NULL,***

***Serial\_Number varchar(600) NOT NULL UNIQUE,***

***Producto\_Disponible boolean DEFAULT FALSE NOT NULL);***

* + ***Creamos la tabla ‘STOCK’ dónde vamos a ver la cantidad de los productos disponibles.***

***DROP TABLE IF EXISTS Stock;***

***CREATE TABLE Stock (***

***ID\_Stock int NOT NULL PRIMARY KEY AUTO\_INCREMENT,***

***Producto\_ID int NOT NULL,***

***Cantidad int UNSIGNED NOT NULL);***

* + ***Creamos la tabla ‘RECIBO’ dónde se guardarán los datos de los productos vendidos a nuestros clientes.***

***DROP TABLE IF EXISTS Recibo***

***CREATE TABLE Recibo (***

***ID\_Recibo int NOT NULL AUTO\_INCREMENTPRIMARY KEY,***

***Producto\_ID int NOT NULL,***

***Provincias\_ID int NOT NULL ,***

***Cliente\_ID int NOT NULL ,***

***Localidad\_ID int NOT NULL);***

* + ***Creamos la tabla ‘GARANTÍA’ dónde vamos a ver que compras tienen la garantía vigente.***

***DROP TABLE IF EXISTS Garantia;***

***CREATE TABLE (***

***ID\_Garanita int NOT NULL PRIMARY KEY AUTO\_INCREMENT,***

***Recibo\_ID int NOT NULL UNIQUE,***

***Producto\_ID int NOT NULL,***

***Provincias\_ID int NOT NULL,***

***Cliente\_ID int NOT NULL ,***

***Localidad\_ID int NOT NULL,***

***Fecha\_de\_Compra datetime DEFAULT NULL ,***

***Inicio\_Garantia datetime DEFAULT NULL ,***

***Fin\_garantia datetime DEFAULT NULL );***

* + ***Creamos la tabla ‘DIRECCIONES’ dónde vamos a anotar la dirección de los clientes y su código postal.***

***DROP TABLE IF EXISTS Direcciones;***

***CREATE TABLE Direcciones(***

***ID\_Direccion int NOT NULL AUTO\_INCREMENT PRIMARY KEY,***

***Cliente\_ID int NOT NULL UNIQUE,***

***Direccion varchar(600) NOT NULL,***

***CP int NOT null);***

* + ***Creamos la tabla ‘ENVÍO’ dónde vamos a dejar acentuado para que cliente, cual dirección, cual producto y que día se hace el envío.***

***DROP TABLE IF EXISTS Envio;***

***CREATE TABLE Envio (***

***ID\_Envio int AUTO\_INCREMENT NOT NULL PRIMARY KEY,***

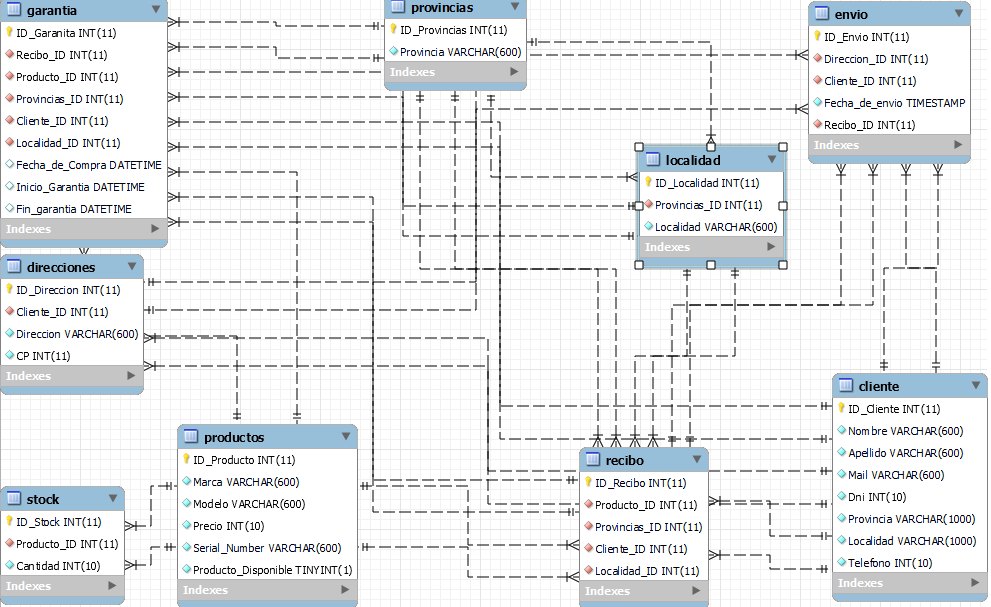
***Direccion\_ID int NOT NULL,***

***Cliente\_ID int NOT NULL ,***

***Fecha\_de\_envio datetime NOT NULL,***

***Recibo\_ID int NOT NULL UNIQUE );***

***Diagrama entidad relación***

******

***Se crea las tablas de relación***

***de uno a muchos***

***1\_ALTER TABLE Localidad***

***ADD FOREIGN KEY(Provincias\_ID)***

***REFERENCES Provincias(ID\_Provincias);***

***2\_ALTER TABLE Stock***

***ADD FOREIGN KEY(Producto\_ID)***

***REFERENCES Productos(ID\_Producto);***

***3\_ALTER TABLE Direcciones***

***ADD FOREIGN KEY(Cliente\_ID)***

***REFERENCES Cliente(ID\_Cliente);***

***4\_ALTER TABLE Envio***

***ADD FOREIGN KEY(Direccion\_ID)***

***REFERENCES Direcciones(ID\_Direccion);***

***5\_ALTER TABLE Envio***

***ADD FOREIGN KEY (Cliente\_ID)***

***REFERENCES Cliente(ID\_Cliente);***

***6\_ALTER TABLE Recibo***

***ADD FOREIGN KEY (Producto\_ID)***

***REFERENCES Productos(ID\_Producto);***

***7\_ALTER TABLE Recibo***

***ADD FOREIGN KEY (Provincias\_ID)***

***REFERENCES provincias(ID\_Provincias);***

***8\_ALTER TABLE Recibo***

***ADD FOREIGN KEY(Cliente\_ID)***

***REFERENCES Cliente(ID\_Cliente);***

***9\_ALTER TABLE Recibo***

***ADD FOREIGN KEY (Localidad\_ID)***

***REFERENCES Localidad(ID\_Localidad);***

***10\_ALTER TABLE Garantia***

***ADD FOREIGN KEY(Recibo\_ID)***

***REFERENCES Recibo(ID\_Recibo);***

***11\_ALTER TABLE Garantia***

***ADD FOREIGN KEY(Producto\_ID)***

***REFERENCES Productos(ID\_Producto);***

***12\_ALTER TABLE Garantia***

***ADD FOREIGN KEY(Provincias\_ID)***

***REFERENCES Provincias(ID\_Provincias);***

***13\_ALTER TABLE Garantia***

***ADD FOREIGN KEY(Cliente\_ID)***

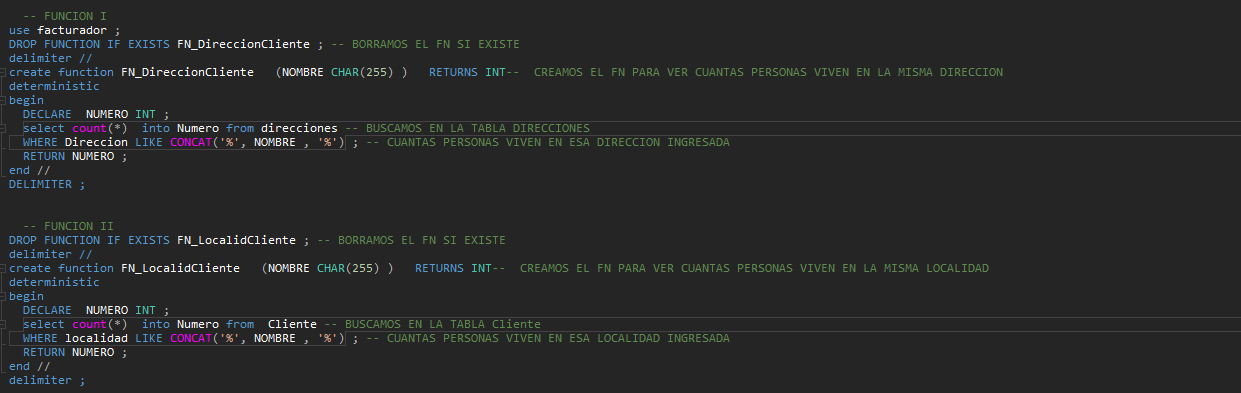
***REFERENCES Cliente(ID\_Cliente);***

***14\_ALTER TABLE Garantia***

***ADD FOREIGN KEY(Localidad\_ID)***

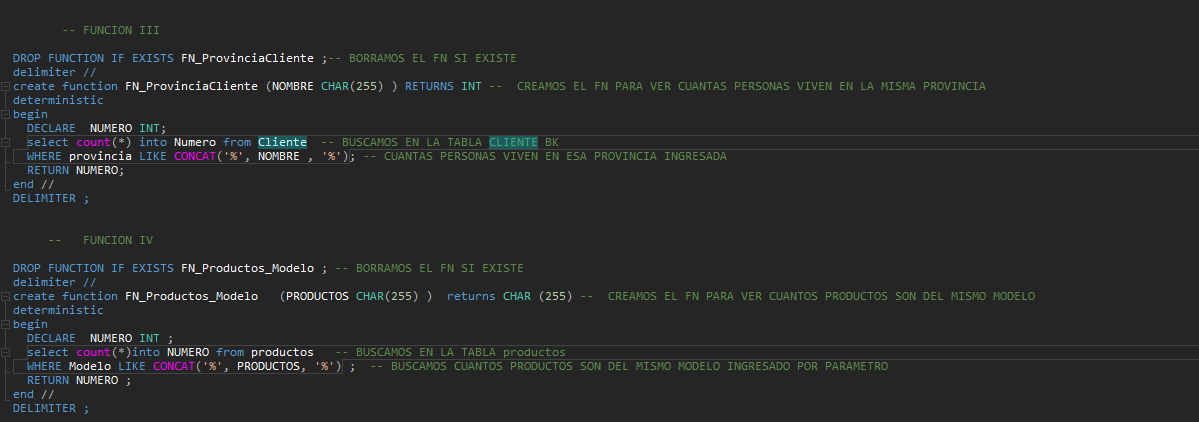
***REFERENCES Localidad(ID\_Localidad);***

***CREACION Y USO DE FUNCIONES***

******

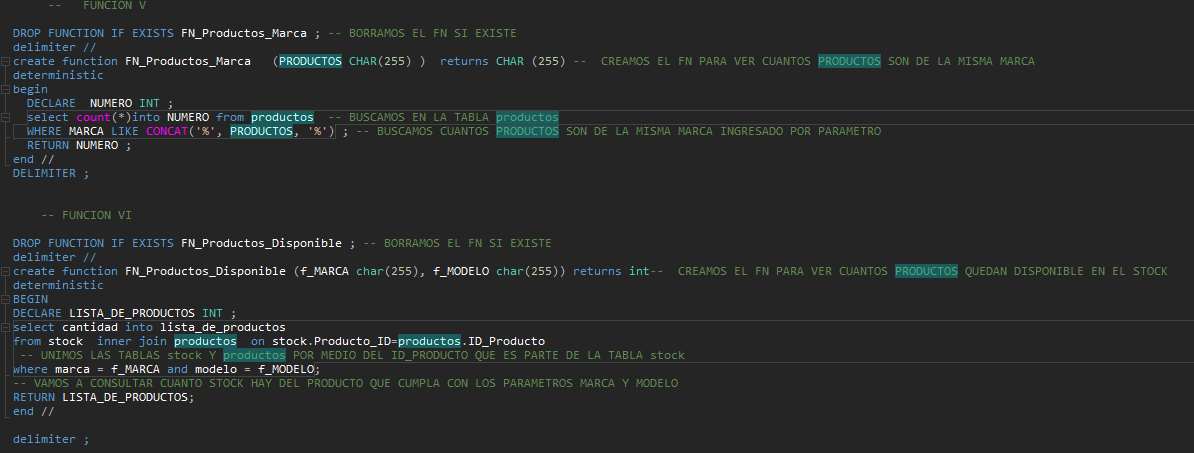
*Función I: Lo que hacemos en la función 1 es buscar cuantas personas/clientes viven en la misma dirección ingresada por parámetro, esto lo busca en la tabla direcciones y nos retorna la unidad de personas viviendo en la dirección ingresada.*

*Función II: Lo que hacemos en la función 2 es buscar cuantas personas/clientes viven en la localidad ingresada por parámetro, esto lo busca en la tabla Cliente y nos retorna la unidad de personas viviendo en la localidad ingresada.*

******

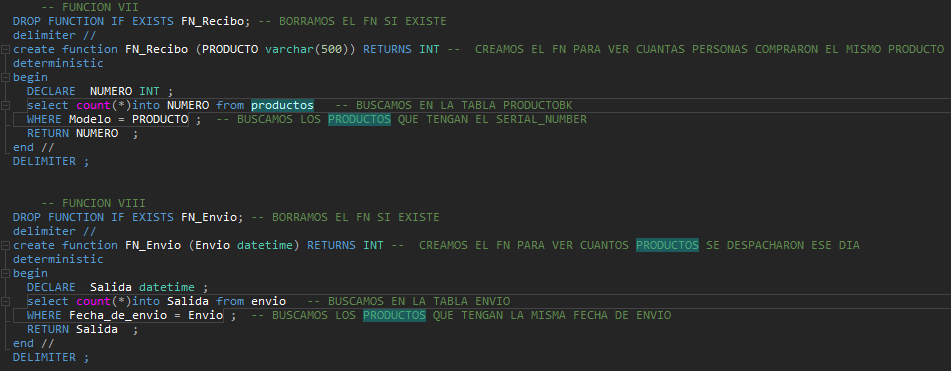
*Función III: Lo que hacemos en la función 3 es buscar cuantas personas/clientes viven en la Provincia ingresada por parámetro, esto lo busca en la tabla Cliente y nos retorna la unidad de personas viviendo en la Provincia ingresada.*

*Función IV Lo que hacemos en la función 4 es buscar cuantos productos son del mismo modelo ingresado por parámetro esto lo busca en la tabla productos y nos retorna la unidad de productos que cumplan con el modelo ingresado*

**

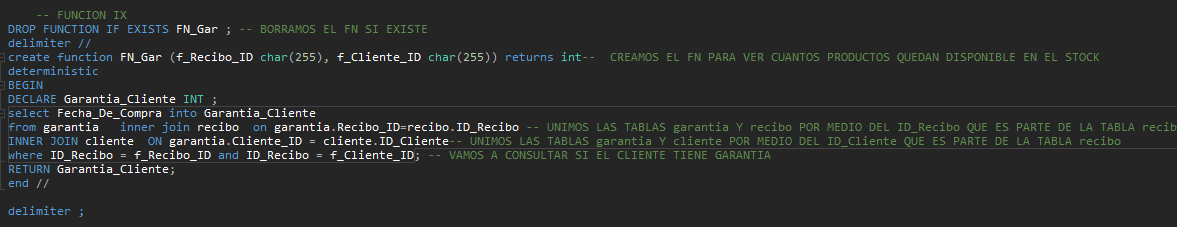
*Función V: Lo que hacemos en la función 5 es buscar cuantos productos tienen la misma marca ingresada por parámetro esto lo busca en la tabla productos y nos retorna la unidad de productos que cumplan con la marca ingresado*

*Función VI: Lo que hacemos en la función 6 es buscar cuantos productos están disponibles que cumplan con los parámetros ingresados, para que esto funcione tenemos que unir la tabla STOCK y la tabla PRODUCTOS por medio del PRODUCTO\_ID que es parte de la tabla STOCK después ingresamos la marca y el modelo que son los parámetros a tener en cuenta y me consulta cuantos productos que cumplan esos parámetros están disponibles en el stock.*

**

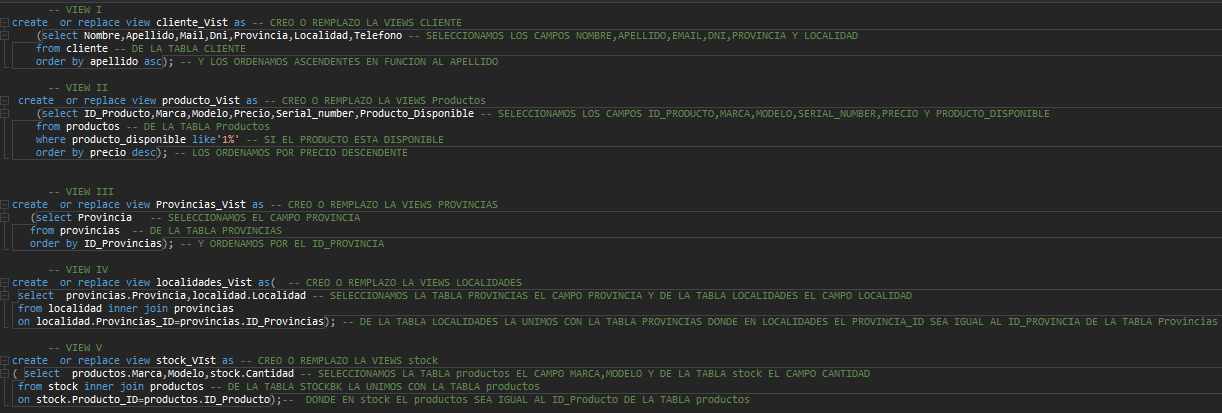
*Función VII: Lo que hacemos en la función 7 es buscar cuantos productos iguales se vendieron para esto consultamos en la tabla PRODUCTOS el parámetro ingresado que va a ser igual al campo Modelo y así nos va a decir la cantidad de veces que se vendió ese producto.*

*Función VIII: Lo que hacemos en la función 8 es buscar cuantos productos fueron despachados en la misma fecha este se va a ingresar por parámetro y nos va a dar el total de productos enviados*

**

*Función IX: Lo que hacemos en la función 9 vamos a consultar si el cliente todavía tiene garantía del producto ingresado por parámetro.*

***CREACION Y USO DE VIEWS***



*Views I: Acá se seleccionan los campos nombres, apellido, mail, DNI, provincia, localidad y Teléfono de la tabla CLIENTE nos muestran todos los datos y los ordenamos ascendentes en función al apellido.*

*Views II: Se seleccionan los Campos id\_producto, marca, modelo, serial\_number, precio y producto\_disponible de la tabla PRODUCTOS, filtramos los productos que estén disponibles y los ordenan en precio descendente.*

*Views III: Seleccionamos el campo provincia de la tabla PROVINCIAS nos muestra sus nombres y los ordena en base a su ID, en forma descendente.*

*Views IV: Lo que hacemos es que se selecciona el campo provincias y localidades de sus tablas y se muestra tal cual correspondan.*

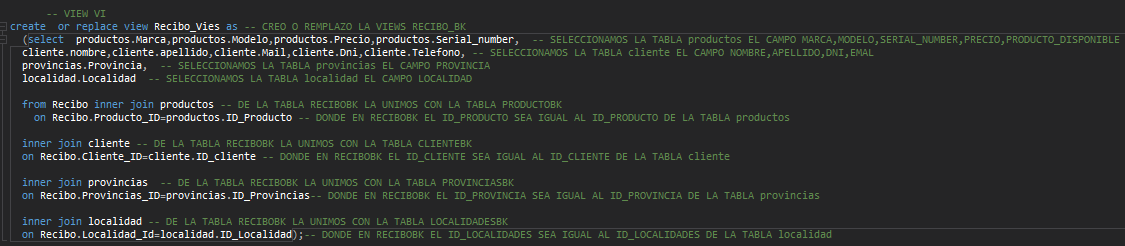
*INNER JOIN*

*Unimos las tablas LOCALIDADES con la tabla PROVINCIAS donde en la tabla principal el campo PROVINCIA\_ID sea igual al ID\_PROVINCIA de la tabla PROVINCIAS.*

*Views V: Se selecciona el campo marca, modelo de la tabla PRODUCTO y cantidad de la tabla STOCK. Y se muesta la cantidad de productos que coincidan en marca y modelo.*

*INNER JOIN*

*Unimos las tablas SOTCK con la tabla PRODUCTO donde en la tabla principal el campo PRODUCTO\_ID sea igual al ID\_ PRODUCTO de la tabla PRODUCTO nos muestra la marca, el modelo y la cantidad de los productos.*



*Views VI: Lo que hacemos es seleccionar el campo marca, modelo, Serial\_number, precio, producto\_disponible de la tabla PRODUCTO, nombre, apellido, DNI, mail, teléfono de la tabla CLIENTE, provincia de la tabla PROVINCAS, localidad de la tabla LOCALIDAD y se muestra un recibo con todos estos datos.*

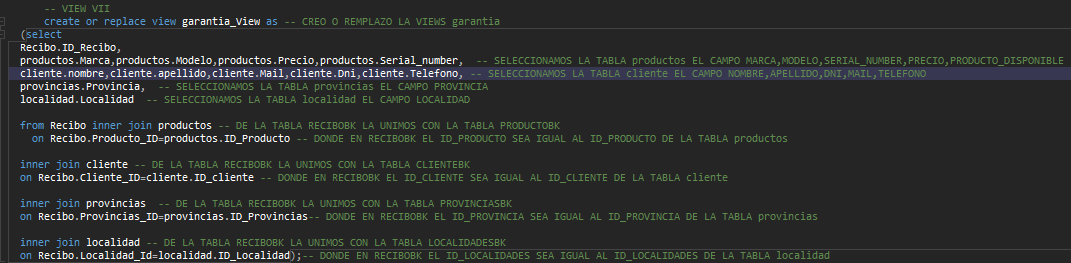
*INNER JOIN*

*Unimos las tablas RECIBO con la tabla PRODUCTO donde en la tabla principal el campo PRODUCTO\_ID sea igual al ID\_ PRODUCTO de la tabla PRODUCTO nos muestra la marca, el modelo, el precio, el serial\_number y la cantidad de los productos.*

*Unimos las tablas RECIBO con la tabla CLIENTE donde en la tabla principal el campo CLIENTE\_ID sea igual al ID\_ CLIENTE de la tabla CLIENTE nos muestra nombre, apellido, DNI, Teléfono y email.*

*Unimos las tablas RECIBO con la tabla PROVINCIAS donde en la tabla principal el campo PROVINCIA\_ID sea igual al ID\_PROVINCIA de la tabla PROVINCIAS nos muestra la provincia del cliente.*

*Unimos las tablas RECIBO con la tabla LOCALIDADES donde en la tabla principal el campo LOCALIDAD\_ID sea igual al ID\_LOCALIDAD de la tabla LOCALIDAD nos muestra la localidad del cliente.*

**

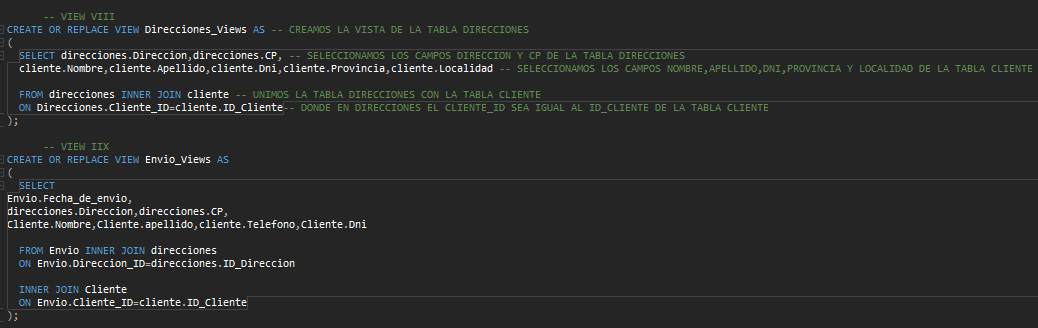
*Views VII: Lo que hacemos es seleccionar el campo id\_recibo de la tabla RECIBO nos va a mostrar el id del recibo para la garantía, marca, modelo, Serial\_number, precio, producto\_disponible de la tabla PRODUCTO, nombre, apellido, DNI, mail, Teléfono de la tabla CLIENTE, provincia de la tabla PROVINCAS, localidad de la tabla LOCALIDAD.*

*INNER JOIN*

*Unimos las tablas RECIBO con la tabla PRODUCTO donde en la tabla principal el campo PRODUCTO\_ID sea igual al ID\_ PRODUCTO de la tabla PRODUCTO nos muestra la marca, el modelo, el precio, el serial\_number y la cantidad de los productos.*

*Unimos las tablas RECIBO con la tabla CLIENTE donde en la tabla principal el campo CLIENTE\_ID sea igual al ID\_ CLIENTE de la tabla CLIENTE nos muestra nombre, apellido, DNI, Teléfono y email.*

*Unimos las tablas RECIBO con la tabla PROVINCIAS donde en la tabla principal el campo PROVINCIA\_ID sea igual al ID\_PROVINCIA de la tabla PROVINCIAS nos muestra la provincia del cliente.*

*Unimos las tablas RECIBO con la tabla LOCALIDADES donde en la tabla principal el campo LOCALIDAD\_ID sea igual al ID\_LOCALIDAD de la tabla LOCALIDAD nos muestra la localidad del cliente.* **

*Views VIII: Lo que hacemos es seleccionar los campos Dirección y CP de la tabla Direcciones y de la tabla Cliente seleccionamos Nombre, Apellido, DNI, Provincia y Localidad, para verificar a donde se tiene que enviar el producto.*

*INNER JOIN*

*Unimos las tablas Direcciones con la tabla Cliente donde en Direcciones el CLIENTE\_ID sea igual al ID\_CLIENTE de la tabla Cliente.*

*Views IX: Lo que hacemos es seleccionar los campos Fecha\_de\_envio de la tabla ENVIO, Dirección, CP De la tabla Direcciones y Nombre, Apellido, Teléfono y DNI de la tabla Cliente, una vez este terminado esto, se realiza el envío.*

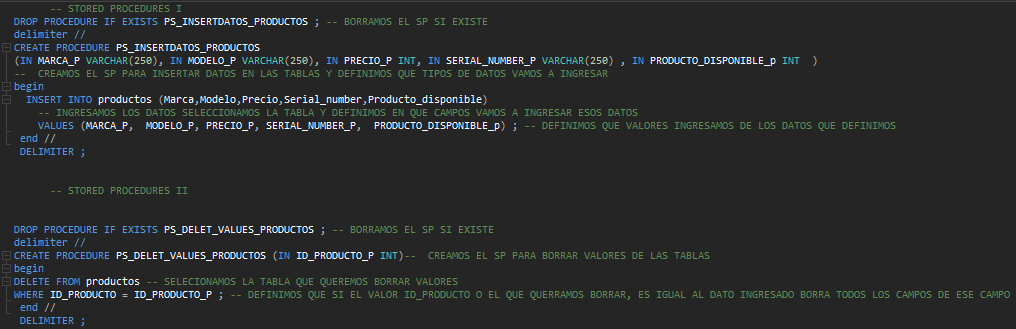
*INNER JOIN*

*Unimos las tablas ENVIO con la tabla Direcciones donde en ENVIO el Direccion\_ID sea igual al ID\_Direccion de la tabla Direcciones*

*Unimos la tabla Envió con la tabla Cliente donde en ENVIO el Cliente\_ID sea igual al ID\_CLIENTE de la tabla Cliente*

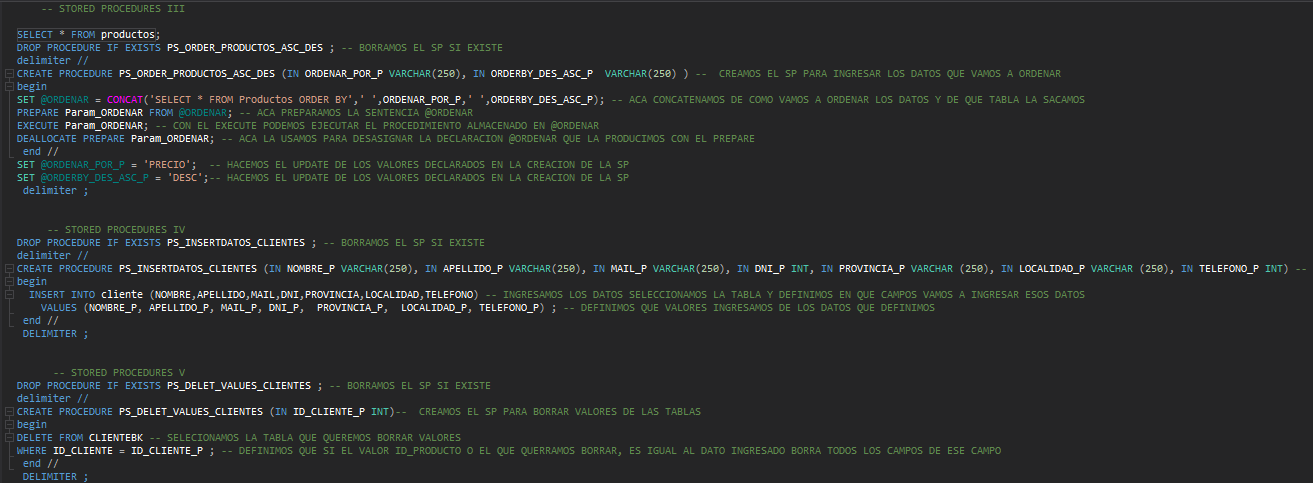
***CREACION Y USO DE LOS***

***STORE PROCEDURES***

******

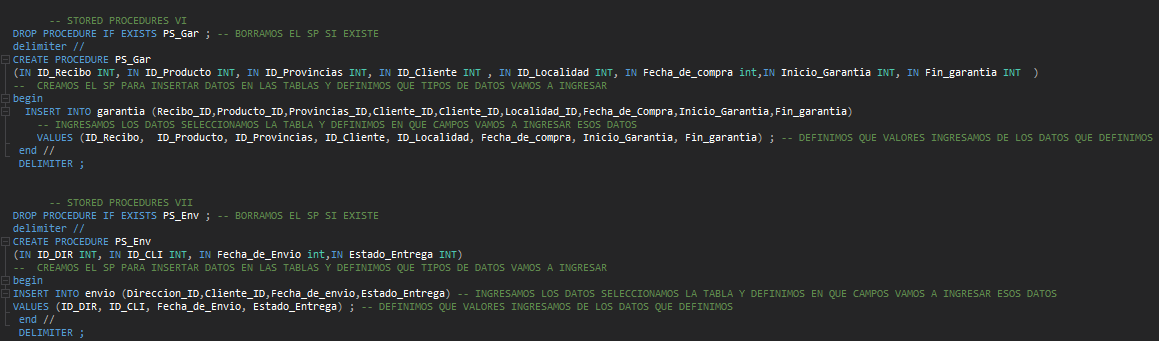
*Stored Produceres I: Lo que el SP 1 hace es que podamos ingresar datos en la tabla elegida, siempre y cuando concuerden con los campos seleccionados por ejemplo elegimos la tabla PRODUCTO y llenamos los datos en el mismo orden de los campos.*

*Stored Produceres II: Lo que el SP 2 hace es que podamos borrar datos en la tabla seleccionada, siempre y cuando el ID del producto ingresado concuerde con el ID del producto en la tabla PRODUCTO.*

*******Stored Produceres III: Lo que el SP 3 hace es ordenar los productos en forma ascendente o descendente de la tabla PRODUCTO así lo decidamos al ingresar el producto y el orden en el que lo queremos ver.*

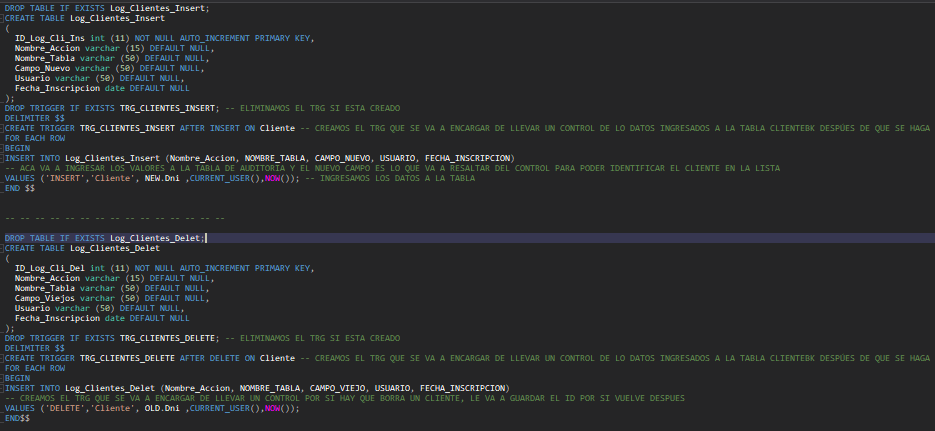
*Stored Produceres IV: Lo que el SP 4 hace es que podamos ingresar datos en la tabla seleccionada, siempre y cuando concuerden con los campos seleccionados por ejemplo elegimos la tabla CLIENTE y llenamos los datos en el mismo orden de los campos.*

*Stored Produceres V: Lo que el SP 5 hace es que podamos borrar datos en la tabla seleccionada, siempre y cuando el ID del cliente ingresado concuerde con el ID del cliente en la tabla CLIENTE*

**

*Stored Produceres VI: Lo que el SP 6 hace es que podamos ingresar datos en la tabla seleccionada, siempre y cuando concuerden con los campos seleccionados por ejemplo seleccionamos la tabla GARANTIA y llenamos los datos en el mismo orden de los campos.*

*Stored Produceres VII: Lo que el SP 7 hace es que podamos ingresar datos en la tabla seleccionada, siempre y cuando concuerden con los campos seleccionados por ejemplo seleccionamos la tabla ENVIO y llenamos los datos en el mismo orden de los campos.****TRIGGERSS***

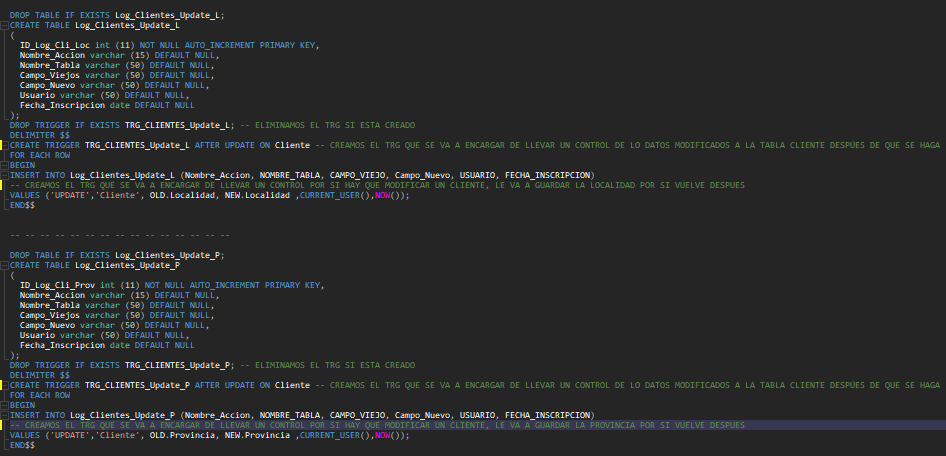
******

*TRIGGER I: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar los clientes en la tabla Cliente y el Nuevo campo es el DNI que es lo que lo identifica como único*

*TRIGGER II: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Borradas a la tabla seleccionada y con el trigger hacemos que después de borrar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser DELET, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es borrar los clientes en la tabla Cliente y el viejo campo es el DNI que es lo que lo identifica como único*

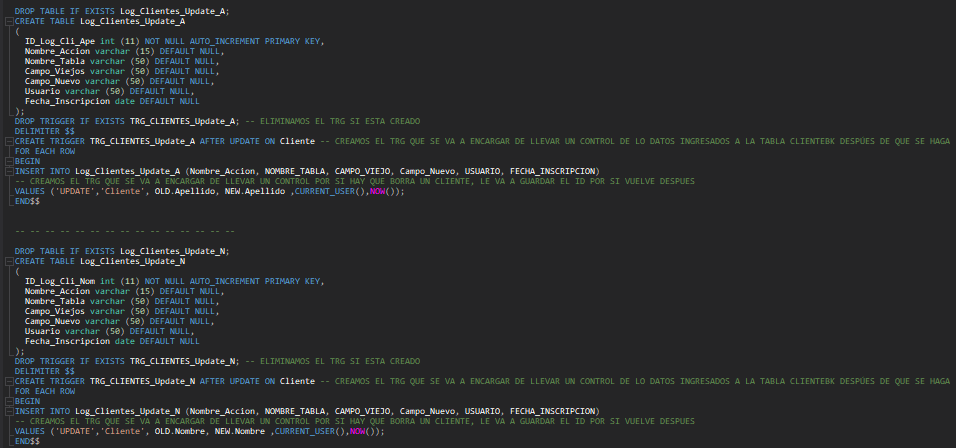
**

*TRIGGER III: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar los clientes en la tabla Cliente y el viejo campo es LA LOCALIDAD y el campo nuevo es LA LOCALIDAD.*

*TRIGGER IV: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar los clientes en la tabla Cliente y el viejo campo es la PROVINCIA y el campo nuevo es la PROVINCIA.*

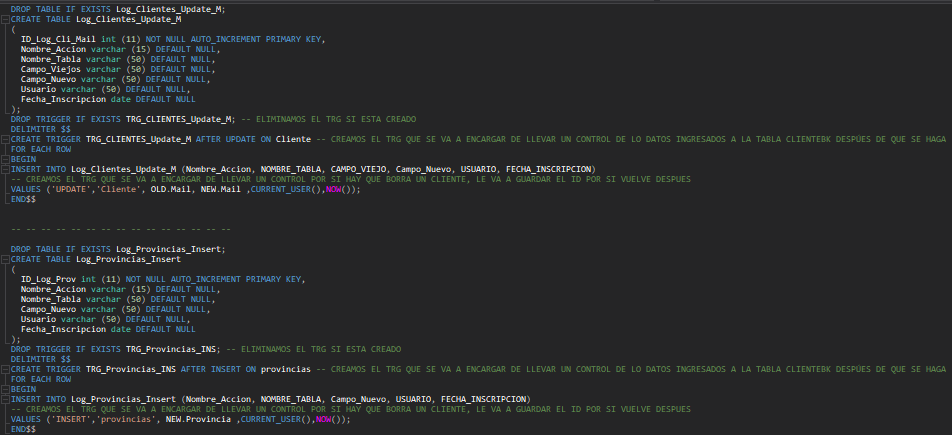
**

*TRIGGER V: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar los clientes en la tabla Cliente y el viejo campo es el Apellido y el campo nuevo es el Apellido.*

*TRIGGER VI: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar los clientes en la tabla Cliente y el viejo campo es el Nombre y el campo nuevo es el Nombre.*

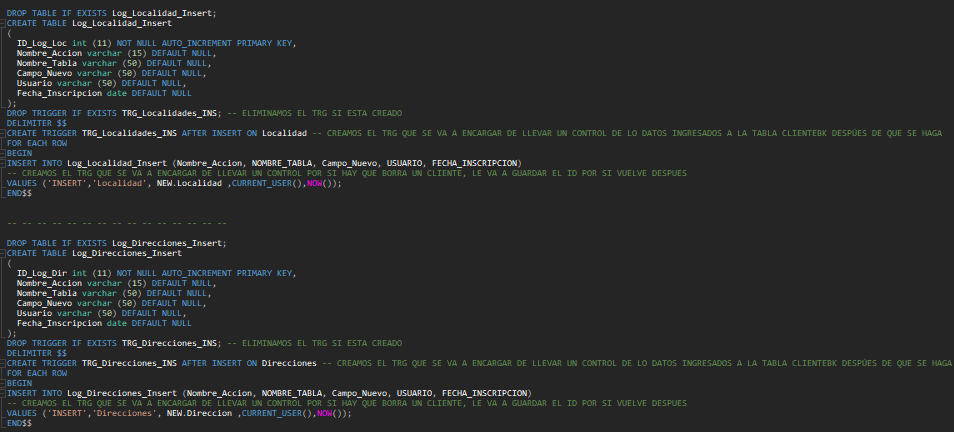
**

*TRIGGER VII: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Cliente guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar los clientes en la tabla Cliente y el viejo campo es el Mail y el campo nuevo es el Mail que es lo que lo identifica como único*

*TRIGGER VIII: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla PROVINCIAS guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar las Provincias en la tabla PROVINCIAS y el Nuevo campo es el nombre de la provincia*

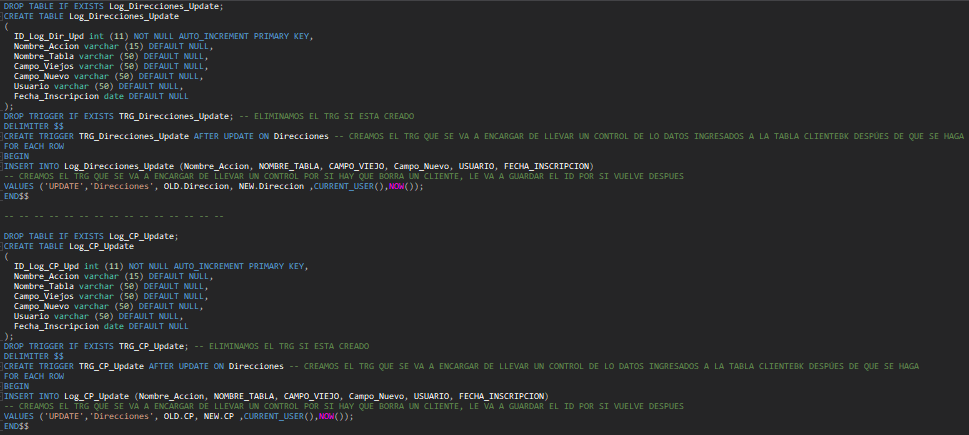
**

*TRIGGER IX: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla LOCALIDAD guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar las Localidades en la tabla LOCALIDAD y el Nuevo campo es el nombre de la Localidades*

*TRIGGER X: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla DIrecciones guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar las Direcciones en la tabla Direcciones y el Nuevo campo es la Dirección*

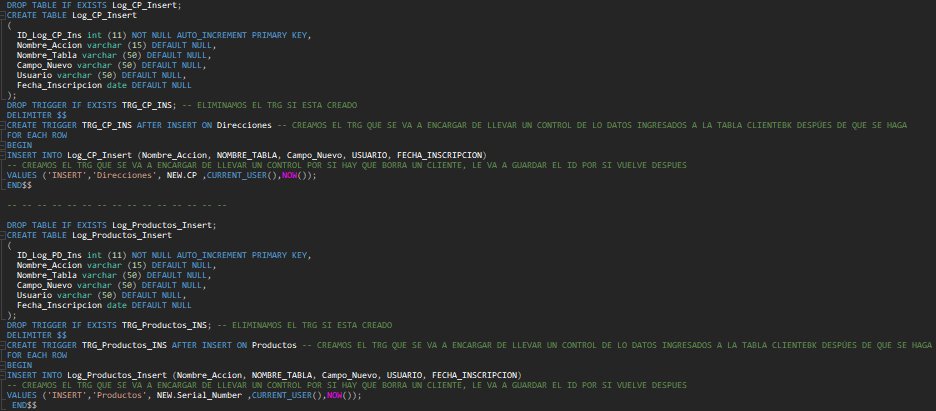
**

*TRIGGER XI: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Direcciones guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar las direcciones en la tabla Direcciones y el viejo campo es la Dirección anterior y el campo nuevo es la Dirección nueva .*

*TRIGGER XII: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Modificadas a la tabla seleccionada y con el trigger hacemos que después de Modificar datos en la tabla Direcciones guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser UPDATE y el campo nuevo en caso de ser UPDATE, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es Modificar los CP en la tabla Direcciones y el viejo campo es el CP anterior y el campo nuevo es el CP nuevo .*

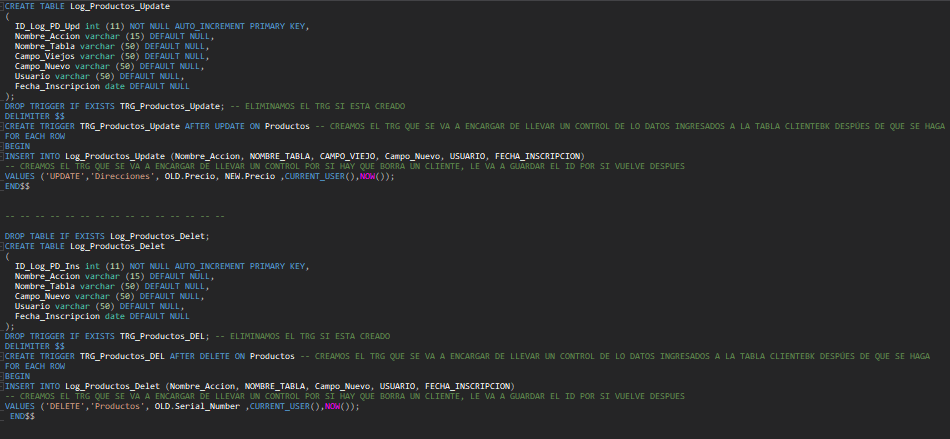
**

*TRIGGER XIII: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla Direcciones guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar los nuevos CP en la tabla Direcciones y el Nuevo campo es el* *CP.*

*TRIGGER XIV: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla PRODUCTOS guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar los nuevos productos en la tabla PRODUCTOS y el Nuevo campo es el Serial\_Number que es lo que lo hace único e identificable.*

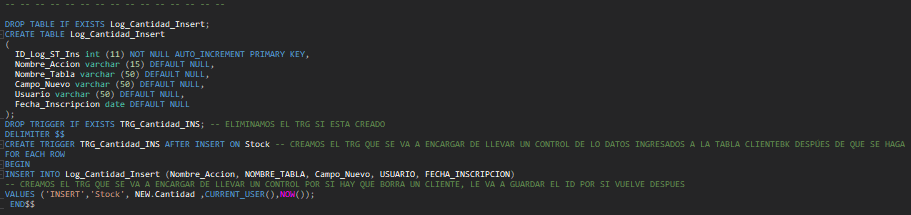
**

*TRIGGER XV: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla PRODUCTOS guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar los nuevos precios en la tabla PRODUCTOS y el Nuevo campo es el precio.*

*TRIGGER XVI: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas Borradas a la tabla seleccionada y con el trigger hacemos que después de borrar datos en la tabla PRODUCTOS guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo viejo en caso de ser DELET, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es borrar los productos vendidos en la tabla PRODUCTOS y el viejo campo es el Serial\_Number que es lo que lo identifica como único*

**

*TRIGGER XVII: Lo que hacemos primero crear una tabla de LOGS que va a llevar el registro de las cosas ingresadas a la tabla seleccionada y con el trigger hacemos que después de insertar datos en la tabla Stock guarde en la tabla LOGS creada recientemente el nombre de la acción(insert,update,delete), el nombre de la tabla(Donde ingresamos esos datos),El campo Nuevo en caso de ser insert, el usuario que lo ingreso que es el que usamos en la db en este caso ROOT, y la fecha de inscripción.*

*En este trigger lo que hacemos es ingresar los nuevos productos en la tabla Stock y el Nuevo campo es la cantidad.*